Detection and Prediction of Errors in EPC Business Process Models

Jan Mendling
Institute of Information Systems and New Media
Vienna University of Economics and Business Administration (WU Wien)
Austria
jan.mendling@wu-wien.ac.at

22th of May 2007

A dissertation submitted to the
Institute of Information Systems and New Media,
Vienna University of Economics and Business Administration (WU Wien).

* * *

Dissertation

Detection and Prediction of Errors in
EPC Business Process Models

Submitted to the Institute of Information Systems and New Media,
Vienna University of Economics and Business Administration, by

Jan
Mendling

in partial fulfillment of the requirements
for the degree of Doctor of Social Sciences and Economics
Under supervision of

Prof. Dr. Gustaf Neumann
Institute of Information Systems
and New Media
WU Wien

Prof. Dr. Wil M. P. van der Aalst
Department of Mathematics
and Computer Science
TU Eindhoven

Abstract

Business process modeling plays an important role in the management of business processes. As valuable design artifacts, business process models are subject to quality considerations. In this context, the absence of formal errors, such as deadlocks, is of paramount importance for the subsequent implementation of the process. This doctoral thesis provides a fourfold contribution to the understanding of such errors in business process models with a particular focus on Event-driven Process Chains (EPCs), a business process modeling language that is frequently used in practice. Firstly, we formalize the operational semantics of EPCs in a novel way so that matching OR-splits and OR-joins never deadlock. Secondly, and based on these semantics, we introduce a soundness criterion for EPCs that offers a precise identification of those models which have errors. For the verification of this soundness notion in practice, we present two analysis tools, a ProM plug-in for a verification based on the reachability graph, and a batch program called *xoEPC* for a verification based on reduction rules. Thirdly, we define a set of business process model metrics that are supposed to serve as predictors for error probability of an individual EPC. Fourthly, we use statistical methods and a sample of about 2000 EPCs from practice to derive a regression function for the prediction of error probability. This function is validated against a holdout set of 113 EPCs from textbooks showing that 90% of the EPCs could be classified correctly as having errors or not. These results emphasize the importance of quality issues in business process modeling and provides the foundations for innovations in tool support.

Zusammenfassung

Geschäftsprozessmodellierung spielt eine wichtige Rolle für das Management von Geschäftsprozessen. Als wertvolle Designartefakte sind Geschäftsprozessmodelle Gegenstand von Qualitätsbetrachtungen. In diesem Zusammenhang ist es von besonderer Wichtigkeit für die nachfolgende Implementierung des Prozesses, dass keine formalen Fehler wie etwa Verklemmungen in den Modellen vorhanden sind. Diese Doktorarbeit liefert vier wesentliche Beiträge zum Verständnis solcher Fehler in Geschäftsprozessmodellen. Das Augenmerk wird insbesondere auf Ereignisgesteuerte Prozessketten (EPKs) gelegt, da diese in der Praxis häufig benutzt werden. Zum Ersten formalisieren wir die operationale Semantik der EPK auf eine neue Art und Weise, sodass zusammengehörende ODER-Verzweigungen und ODER-Zusammenführungen niemals verklemmen. Zum Zweiten, und darauf aufbauend, stellen wir ein Korrektheitkriterium für EPKs vor, das eine präzise Identifikation von solchen Modellen ermöglicht, die Fehler enthalten. Für die praktische Verifizierung dieses Korrektheitskriteriums präsentieren wir

zwei Analysewerkzeuge, zum einen einen ProM-Plug-in zur Verifikation auf Basis des Erreichbarkeitsgraphens, und zudem ein Stapelverarbeitungsprogramm namens *xoEPC* zur Verifikation mit Hilfe von Reduktionsregeln. Zum Dritten definieren wir eine Menge von Geschäftsprozessmodellmetriken, die als Anzeiger für die Fehlerwahrscheinlichkeit einer einzelnen EPK dienen sollen. Zum Vierten benutzen wir statistische Methoden und eine Stichprobe von etwa 2000 EPKs aus der Praxis, um eine Regressionsfunktion zur Vorhersage von Fehlerwahrscheinlichkeiten abzuleiten. Diese Funktion wird anhand einer zweiten Stichprobe von 113 EPKs aus Lehrbüchern validiert, welche zeigt, dass 90% der EPKs richtig als fehlerhaft oder fehlerfrei klassifiziert werden konnten. Die Ergebnisse unterstreichen die Wichtigkeit von Qualitätsbetrachtungen in der Geschäftsprozessmodellierung und bieten eine Grundlage für Innovationen in der Werkzeugunterstützung.

Acknowledgement

On these pages it is time to thank all the people who were involved in this thesis, but who are not an author. I want to start with Prof. Dr. Gustaf Neumann, who served as a first supervisor, and Prof. Dr. Wil van der Aalst who, was second supervisor. In October 2003 I started working at Prof. Neumann's academic entity which is now called Institute of Information Systems and New Media (i.e. Institut für Wirtschaftsinformatik und Neue Medien). Prof. Neumann offered me a creative work environment where I could develop my ideas and academic writing skills. He was open and supportive with all matters, in particular, a change of the topic of my thesis, conference attendances, and research visits abroad. In 2004, I met Prof. van der Aalst at a conference and after several meetings at other conferences, he finally agreed to be my second supervisor. His feedback was extraordinarily helpful for the formalization of my concepts.

Additionally, I want to thank Prof. Dr. Markus Nüttgens for raising my interest in business process modeling and EPCs, and for his continuous support since my diploma thesis in Trier, Germany; Prof. Dr. Walter Schertler and Prof. Dr. Axel Schwickert for supporting my application in Vienna by serving as a reference; Prof. Dr. Carlos de Backer for being a role model in teaching Information Systems and explaining IT concepts in Dutch; and Prof. Dr. Hans Czap for his efforts in establishing the Diplomstudiengang Wirtschaftsinformatik at the University of Trier. Furthermore, I thank all my colleagues at the Institute of Information Systems and New Media and the Institute of Management Information Systems at the Vienna University of Economics and Business Administration (WU Wien) for providing a friendly work environment.

I was happy to collaborate with several excellent people on different paper projects: Wil van der Aalst, Paul Barborka, Alberto Brabenetz, Jorge Cardoso, Boudewijn van Dongen, Michael Hafner, Lukas Helm, Thomas Hornung, Cirano Iochpe, Georg Köldorfer, Agnes Koschmider, Kristian Bisgaard Lassen, Jean Michael Lau, Michael Moser, Michael zur Muehlen, Martin Müller, Gustaf Neumann, Markus Nüttgens, Cristian Pérez de Laborda, Andreas Pinterits, Adrian Price, Michael Rausch, Jan Recker, Manfred Reichert, Hajo Reijers, Michael Rosemann, Frank Rump, Bernd Simon, Carlo Simon, Guido Sommer, Mark Strembeck, Gerald Stermsek, Lucinéia Heloisa Thom, Irene Vanderfeesten, Eric Verbeek, Barbara Weber, Fridolin Wild, Uwe Zdun, and Jörg Ziemann. Thank you for sharing your insights with me. I also thank Kalina Chivarova for gathering some of the EPC models of the holdup sample in her bachelor thesis. Special thanks goes to Herbert Liebl, who had the nice idea to generate SVG graphics from the error analysis and highlight the errors in the EPC models.

Zuletzt einige Worte des Danks auf deutsch. Meinen Eltern danke ich für ihre nimmermüde Unterstützung und ihr Vertrauen in meine Fähigkeiten. Ohne ihren Rückhalt und den meiner Schwester, Omas und Opas, Tanten und Onkels, Ur-Omas und Ur-Opas und meiner Freunde wäre diese Arbeit nicht denkbar. Dies gilt insbesondere für Wiebke, mit der ich gemeinsam nach Wien gegangen bin. Zuletzt danke ich Leni. Sie gab mir die Geborgenheit und Wärme, die ich brauchte, um diese Arbeit zu einem guten Ende zu führen. Ich freue mich auf alles, was wir noch gemeinsam erleben werden.

Jan Mendling
April 2007

Contents

List of Figures

List of Tables

List of Acronyms

ACM	Association for Computing Machinery
AD	Anno Domini
AND	Logical and-operator
ARIS	ARchitecture of Integrated Information Systems
BPEL	Business Process Execution Language for Web Services
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BWW	Bunge-Wand-Weber
CASE	Computer Aided Software Engineering
CC	Cyclomatic complexity, cyclomatic number
CCD	Cyclomatic complexity density
CDIF	CASE Data Interchange Format
C-EPC	Configurable Event-driven Process Chain
CFC	Control-flow complexity
COCOMO	Constructive Cost Model
CPN	Colored Petri Nets
DOM	Document Object Model
ebXML	Electronic Business XML
ECC	Essential cyclomatic complexity
EPC	Event-driven Process Chain
EPK	Ereignisgesteuerte Prozesskette
EPML	Event-driven Process Chain Markup Language
ERD	Entity-relationship diagram
ERP	Enterprise Resource Planning
ET	Entscheidungstabellen-operator
FSM	Finite State Machine
GHz	Giga-Hertz
GI	Gesellschaft für Informatik e.V., German Informatics Society
GoM	Guidelines of Modeling

GQM	Goal-Question-Metric
GXL	Graph Exchange Language
HTML	Hypertext Markup Language
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IFC	Information Flow Complexity
ISO	International Organization for Standardization
IT	Information technology
IWi	Institut für Wirtschaftsinformatik
KIM	Kölner Integrationsmodell
KOPeR	Knowledge-based organizational process redesign
LoC	Lines of Code
LTL	linear temporal logic
MI	Multiple Instantiation
modEPC	Modified Event-driven Process Chain
OASIS	Organization for the Advancement of Structured Information Standards
oEPC	Object-oriented Event-driven Process Chain
OR	Logical or-operator
ProM	Process Mining tool of TU Eindhoven
rEPC	Real-time Event-driven Process Chain
RG	Reachability Graph
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung
SCOOP	System for Computerization of Office Processing
SEQ	Sequence-operator
SNA	Social network analysis
SPSS	Statistical Package for the Social Sciences
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TAM	Technology acceptance model
Tcl	Tool command language
tDOM	Tool command language Document Object Model
UML	Unified Modeling Language
UML AD	Unified Modeling Language Activity Diagram
US	United States
Wf. nets	Workflow nets
WfMC	Workflow Management Coalition
Woflan	WOrkFLow ANalyzer

WS-CDL	Web Services Choreography Description Language
xEPC	Agent-oriented Event-driven Process Chain
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
xoEPC	Extended Object Event-driven Process Chain
XOR	logical exclusive or-operator
XOTcl	Extended Object Tool command language
XPDL	Extensible Markup Language Process Definition Language
YAWL	Yet Another Workflow Language
yEPC	Yet another Event-driven Process Chain

Chapter 1

Introduction

This chapter provides an introduction to this doctoral thesis. After a discussion of the general motivation in Section 1.1, we present the research contribution of this thesis in Section 1.2. In Section 1.3, we discuss the findings from an epistemological point of view and relate them to design science and behavioral science approaches to information systems research. Finally, Section 1.4 closes this chapter with an outlook on the structure of this thesis.

1.1 Motivation

The importance of Business Process Management (BPM) is reflected by the figures of the related industry. For example, Wintergreen Research estimates that the international market for BPM-related software and services accounted for more than 1,000 million US dollars in 2005 with a tendency of rapid growth in the next couple of years [Win06]. Furthermore, the plethora of popular and academic textbooks (e.g. [HC93, Dav93, JB96, Sch98a, ACD$^+$99, Sch00, LR00, ADO00, AH02, MCH03, BKR03, Kha04, LM04, Hav05, SF06, Sta06, Jes06, SCJ06, WLPS06, KRM06, Smi07]) as well as international professional and academic conference series, such as the BPM conference [AHW03, DPW04, ABCC05, DFS06], confirm the importance of BPM. Despite the overall recognition of its importance, several fundamental problems remain unsolved by current approaches.

A particular problem in this context is the lack of research regarding what is to be considered good design. The few contributions in this area reveal an incomplete understanding of quality aspects in this regard. Business process modeling as a sub-discipline of BPM faces a particular problem. Often, modelers who have little background in formal methods, design models without understanding the full implications of their specification (see e.g. [PH07]). As a consequence, process models designed on a business level can hardly be reused on an execution level since they often suffer from formal errors, such as deadlocks.[1] Since the costs of errors increase exponentially over the development life cycle [Moo05], it is of paramount importance to discover errors as early as possible. A large amount of work has been conducted to try to cure the symptoms of this weak understanding by providing formal verification techniques, simulation tools, and animation concepts. Still, several of these approaches cannot be applied since the business process modeling language in use is not specified appropriately. Furthermore, this stream of research does not get to the root of the problem. As long as we do not understand why people introduce errors in a process model, we will hardly be able to improve the design process. There is some evidence on error rates for one particular collection of business process models from practice [MMN$^+$06b, MMN$^+$06c].[2] We will take this research as a starting point to contribute to a deeper understanding of errors in business process models.

[1] In the subsequent chapters, we will distinguish between two major types of errors. Firstly, formal errors can be identified algorithmically with verification techniques. Secondly, inconsistencies between the real-world business process and the process model can only be detected by talking to stakeholders. The focus of this thesis will be on formal errors.

[2] Classroom experiences are reported, for example, in [MSBS02, Car06].

1.2 Research Contributions

The research objective of this doctoral thesis is the development of a framework for the detection of formal errors in business process models, and the prediction of error probability based on quality attributes of these models. We will focus on Event-driven Process Chains (EPCs), a business process modeling language that is heavily used in practice. The advantage of this focus is, firstly, that the results of this thesis are likely to have an impact on current modeling practices. Secondly, there is a large empirical basis for analysis. By tapping the extensive stock of EPC model collections, we aim to bring forth general insights into the connection between process model attributes and error probability. In order to validate such a connection, we first need to establish an understanding of model attributes that are likely connected with error probability. Furthermore, we must formally define an appropriate notion of correctness, which gives an answer to the question whether a model has a formal error or not. It is a prerequisite to answering this question that we define the operational semantics of the process modeling language, i.e. EPCs, in a formal way. Against the state of the art, this thesis provides the following technical contributions.

Formalization of the OR-join: The semantics of the OR-join have been debated for more than 10 years now. Existing formalizations suffer either from a restriction of the EPC syntax (see e.g. [CS94, LA94, LSW98, Aal99, DR01]) or from non-intuitive behavior (see e.g. [NR02, Kin06, AH05, WEAH05]). In Chapter 3 of this thesis we formalize the EPC semantics concept as proposed in [MA06]. In comparison to other approaches, this novel formalization has the advantage that it is not restricted to a subset of EPCs, and that it provides intuitive semantics for blocks of matching OR-splits and joins since they cannot deadlock. The calculation of the reachability graph was implemented as a plug-in for ProM as a proof of concept. In this way, this novel semantics definition contributes to research on the specification of business process modeling languages.

Verification of process models with OR-joins and multiple start and end events: Verification techniques for process models with OR-joins and multiple start and end events suffer from one of two problems. Firstly, they build on an approximation of the actual behavior and, therefore, do not provide a precise answer to the verification problem, e.g. by considering a relaxed notion of soundness [DR01], by involving user decisions [DAV05], or by approximating relaxed soundness with invariants [VA06]. Secondly, there are verification approaches for semantics definitions (see [CFK05, WAHE06]) that suffer from the previously mentioned non-intuitive behavior. While this is not a problem of the verification itself, all these approaches are not tailored to cope with multiple start and end events. In Chapter 4 of this thesis, we specify a dedicated soundness criterion for EPC business process models with OR-joins and multiple start and end events. Furthermore, we define two verification approaches for EPC soundness, one as an explicit analysis of the reachability graph, and a second based on reduction rules to provide a better verification performance. Both approaches were implemented as a proof of concept. In this way, we contribute to the verification of process models with OR-joins and multiple start and end events, and in particular, we extend the set of reduction rules for business process models.

Metrics for business process models: Metrics play an important role in the operationalization of various quality-related aspects in software engineering, network analysis, and business process modeling. Several authors use metrics to capture different aspects of business process models that are presumably related to quality (see e.g. [LY92, Nis98, Mor99, RV04, Car05d, BG05, CGP$^+$05, CMNR06, LG06, ARGP06c, MMN$^+$06b, MMN$^+$06c]). A problem of these works is that business process-specific concepts like sequentiality, decision points, concurrency, or repetition are hardly considered, and too often simple count metrics are defined. Furthermore, there appears to be little awareness of related research, maybe because process model measurement is conducted in separate disciplines including software process management, network analysis, Petri nets theory, and conceptual modeling. In Chapter 5 of this thesis, we will provide an extensive list of metrics for business process models and relate it to previously isolated research. Beyond that, we provide a detailed discussion of the rationale and the limitations of each metric, which is meant to serve as a predictor for error probability. We formulate a hypothesis for each metric based on whether it is positively or negatively correlated with error probability.

Validation of metrics as error predictors: Up to now, there is little empirical evidence for the validity of business process model metrics as predictors for error probability. Some empirical work was conducted, but with a different focus. *Lee and Yoon* investigate the empirical relationship between parameters of Petri nets and their state space [LY90, LY92]. *Canfora et al.* empirically evaluate the suitability of metrics to serve as predictors for maintainability of the process model [CGP$^+$05]. *Cardoso* analyzes the correlation between the control flow complexity metric with the perceived complexity of process models [Car06]. Most related to this thesis is an analysis of the SAP Reference Model where *Mendling et al.* test a set of simple count metrics as error predictors [MMN$^+$06b, MMN$^+$06c]. In Chapter 6 of this thesis, we use logistic regression for the test which is similar to the analysis of the SAP Reference Model. Still, we consider both the broader set of metrics from Chapter 5, a precise notion of EPC soundness as defined in Chapter 4, and a much broarder sample of EPC models from practice. The results do not only show that certain metrics are indeed a good predictor for error probability, but also that simple count metrics fail to capture important aspects of a process model.

Little research in information systems tries to combine design science and behavioral science research paradigms (see e.g. [BH05]). Since the previously listed contributions cover both design and behavioral aspects, we consider the main contribution of this thesis to be the innovative and holistic combination of both these research paradigms in order to deliver a deeper understanding of errors in business process modeling.

1.3 Epistemological Position

This thesis contributes to information systems research as defined by *Hevner, March, Park, and Ram* [HMPR04]. It covers different research aspects that build on both design science and behavioral science paradigms. Section 1.3.1 introduces the Information Systems Research Framework as presented by *Hevner, March, Park, and Ram* [HMPR04], that overarches design science and behavioral science in information systems research.

Section 1.3.2 uses the information systems research guidelines to discuss in how far this thesis meets information systems research standards.

1.3.1 Information Systems Research Framework

Information systems research is the study of phenomena related to information systems. Information systems research and its German counterpart Wirtschaftsinformatik build on both design science and behavioral science research. According to *Hansen and Neumann* [HN05], it is defined as follows: "The study that is concerned with design of computer-based information systems in business is called Wirtschaftsinformatik (in English: Management Information Systems, Information Systems, Business Informatics). It is understood to be an interdisciplinary subject between business science and computer science" (my translation).[3] This definition stresses the design science paradigm which is typical for the European information systems community [BH05, BN07], but it also covers behavioral aspects related to design. Only recently, there has been a trend to widen the scope of Wirtschaftsinformatik by taking advantage of more behavioral, especially empirical, methodologies [BH05].

Behavioral science "seeks to develop and justify theories [...] that explain or predict organizational and human phenomena surrounding the analysis, design, implementation, management, and use of information systems" [HMPR04, p.76]. A typical example of a theory that follows a behavioral science paradigm is the technology acceptance model (TAM) [Dav89]. According to the TAM, user acceptance of information technology can be explained by two major factors: perceived usefulness and perceived ease of use. Since information systems are created by making design decisions, such insights into behavioral aspects provide feedback for the design of new artifacts.

The foundations of information systems research as a *design science* were elaborated in the seminal work of *Simon* on the *Sciences of the Artificial* [Sim96]. In this context, design science is understood as a problem-solving process. A key characteristic of prob-

[3] Similar definitions are given by *Mertens, Bodendorf, König, Picot and Schumann* [MBK+98], *Stahlknecht and Hasenkamp* [SH05], *Ferstl and Sinz* [FS98], *Heinrich, Heinzl, and Roithmayr* [HHR07], or *Lehner* [Leh97].

lems in design science is *wickedness*, i.e. there is no definitive formulation of the problem due to unstable requirements, ill-defined environmental context, complex interactions, inherent change, and of psychological and social factors being involved (cf. [HMPR04]). Therefore, the solution cannot be assessed by truth, but rather by utility. Based on the assumption of bounded rationality of a human as a problem-solver, *Simon* advocates to accept satisficing solutions by designing and creating useful artifacts. In information systems research, design science relates to building and evaluating design artifacts including constructs, models, methods, and instantiations (cf. [MS95]). These artifacts facilitate the exploration of the space of design choices [BBC$^+$04]. Information systems design theories prescribe effective development practices that can be applied for a particular class of user requirements to construct a certain type of system solution [MMG02]. The created information systems artifacts influence and extend the capabilities of organizations and human problem-solving, i.e. they establish a new reality. Respective theories on their application and impact are expected to follow their development and use [HMPR04].

While "behavioral science addresses research through the development and justification of theories that explain or predict phenomena related to the identified business need," design science "addresses research through the building and evaluation of artifacts designed to meet the identified business need. The goal of behavioral-science research is truth. The goal of design-science research is utility" [HMPR04]. The assessment of artifacts (evaluation) or theories (justification) can lead to the identification of weaknesses. Such insight can be used for refinement of artifacts and theories. The research design of this thesis combines both paradigms following the concept of *Hevner et al.* that design and behavioral science are complementary: "truth informs design and utility informs theory" [HMPR04].

A key characteristic of *information systems* in organizations is that they are utilized for "improving the effectiveness and efficiency of that organization" [HMPR04, p.76]. Accordingly, the overall goal of information systems research can be defined as to "further knowledge that aids in the productive application of information technology to human organizations and their management" [ISR02]. Thus, information systems research is conducted in an *environment* that involves people, organizations, and technology in order to enhance the *knowledge base* of foundations and methodologies in this area (cf.

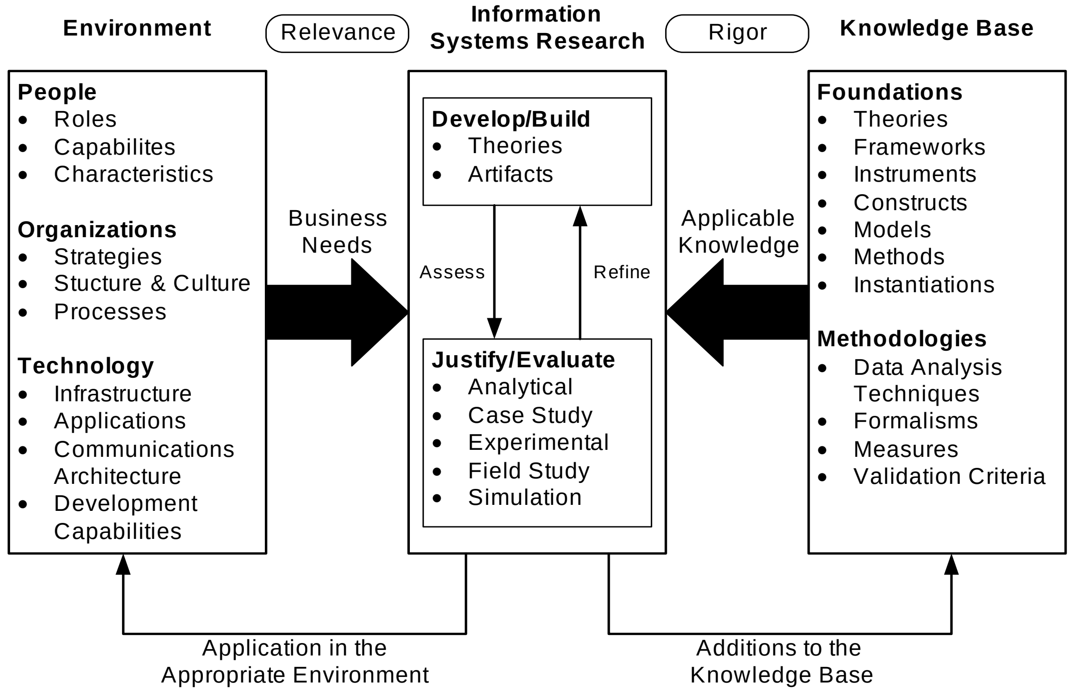


Figure 1.1: Information Systems Research Framework as defined by Hevner et al. [HMPR04, p.80]

Figure 1.1). Due to the involvement of people and organizations, such knowledge can be acquired following two different research paradigms: behavioral science and design science. Both build on a creative activity of developing theories or building artifacts, respectively, and an analytical activity of justification or evaluation, respectively (see Figure 1.1).

The *environment* of information systems research includes those entities that define the problem space, i.e. people, organizations, and technology. It defines the background against which business needs emerge. These business needs are influenced by the roles, capabilities, and characteristics of people, and shaped in consideration of an organization's strategy, structure, culture, and business processes. Moreover, business needs reflect current and prospective technology such as infrastructure, applications, communications architecture, and development capabilities. The researcher aligns her problem perception to these factors in order to establish *relevance*.

The *knowledge base* offers solutions to problems which are already well understood. Development and building can rely on foundations like theories, frameworks, instruments, constructs, models, methods, and instantiations that have resulted from prior research. Methodologies like data analysis techniques, formalisms, measures, and validation criteria are valuable in the justification and evaluation phase. The researcher applies existing foundations and methodologies to a given problem in order to establish *rigor*. Behavioral science often considers empirical evidence, while design sciences tends to use mathematical methods more frequently. The overall goal of both behavioral and design science is to address the business need and to contribute to the knowledge base for future application. The lack of addition to the knowledge base can be used to distinguish routine design and design research. While routine design tackles business needs by applying existing knowledge, design research establishes either innovative solutions to unsolved problems or more efficient or effective solutions to solved problems. Accordingly, design research contributes to the knowledge base while routine design does not.

1.3.2 Relation to Information Systems Research Guidelines

The Information Systems Research Framework emphasizes the similarities between behavioral science and design science. Related to that, *Hevner et al.* suggest a set of seven guidelines for effective information systems research, in particular for works with a design science focus [HMPR04]. On the following pages, we use these guidelines to discuss in how far this thesis meets information systems research standards.

Guideline 1: Design an Artifact Information systems research aims to design purposeful artifacts addressing business needs within an organizational setting. Artifacts in this context include constructs, models, methods, and instantiations [HMPR04]. In this thesis, we formalize EPC semantics, EPC soundness, and error metrics as constructs that can be used to analyze and simulate EPC business process models. Furthermore, we define methods in this sense to calculate the reachability graph, to verify soundness based on reachability graph analysis as well as reduction rules, and to calculate metrics from the process models. Finally, we present prototypical implementations of these methods (i.e. instantiations) as a plug-in for ProM and as a software component called xoEPC in order to demonstrate feasibility.

Guideline 2: Problem Relevance Relevance of information systems research is constituted by addressing a problem of development or practical application of information systems; and in particular, their planning, management, design, operation, and evaluation [HMPR04]. The general business need of this research stems from a wide-spread application of business process management in practice, and of EPCs as a modeling language in particular. The findings and concepts presented in this thesis contribute to several aspects of quality assurance in business process modeling.

Guideline 3: Design Evaluation The utility of an artifact in a given problem situation must be clearly established using evaluation methods [HMPR04]. The completeness and the correctness of the EPC semantics definition and the soundness analysis is checked using analytical methods. The usefulness of business process model metrics is first evaluated in a descriptive way before using statistical methods. The implementations of the verification methods were extensively tested with numerous EPC models.

Guideline 4: Research Contribution The design research has to provide a novel, significant, and general contribution to the knowledge base; otherwise it has to be considered as design routine [HMPR04]. The contributions have already been presented in Section 1.2. They include a novel formalization of the OR-join (design science), two verification approaches for process models with OR-joins and multiple start and end events (design science), metrics for business process models (design science), and a validation of the metrics as predictors of error probability using an extensive set of EPC business process models from practice (behavioral science).

Guideline 5: Research Rigor Rigor refers to the way in which construction and evaluation of design science is conducted. This implies that the researcher has to effectively make use of the knowledge base and its methodologies and foundations [HMPR04]. For this thesis, we took advantage of prior research on business process modeling languages, predicate logic, formal semantics, graph theory, software measurement, and logistic regression.

Guideline 6: Design as a Search Process Problem solving in design science can be defined as utilizing suitable means to reach desired ends while respecting laws imposed by the environment [Sim96]. Suitable means in this context refer to an available operation that can be used to build a solution, ends represent goals and constraints, and laws capture forces of the environment that cannot be controlled. The wickedness of the design-science problem implies that means, ends, and laws cannot be represented on the level of completeness and precision that would be needed for an optimization problem. The problem of finding predictors for error probability in business process models exactly displays this wickedness. In this thesis, we seek to establish a satisficing solution in the terms of *Simon*, based on a set of business process model metrics and on a notion of correctness called EPC soundness. In this setting, it is crucial to demonstrate that a certain solution *does* work, even if it is not yet completely understood *why* it works (cf. [HMPR04]). Using a logistic regression approach, we are not only able to show that this set of metrics does suit for predicting errors, but also that the hypothetical direction of the impact can be validated and that it outperforms existing approaches.

Guideline 7: Communication of Research The design solution has to be presented to both the academic community and to practitioners who might be interested in the findings [HMPR04]. For the research community, communication extends the knowledge base and offers repetition of research in order to check for correctness. Working on this thesis has led to the publication of five journal articles, five book chapters, 49 workshop and conference papers, and 19 technical reports and popular publications. This means that several concepts of this thesis are already publicly available as part of the information systems knowledge base.

Relating this thesis to the information systems research guidelines highlights that it suffices international research standards in this discipline and that it enhances its knowledge base in several directions.

1.4 Structure of this Thesis

This thesis is organized in seven chapters. It starts with a general overview of business process management, continues with semantics of Event-driven Process Chains and the verification of soundness before discussing metrics for business process models that are subsequently validated for their capability to predict error probability.

Chapter 1: Introduction In this chapter, we sketch the motivation of this thesis, present its contributions, and discuss its epistemological position related to information systems research.

Chapter 2: Business Process Management This chapter discusses the backgrounds of business process management and defines important terms related to it. Furthermore, it sketches the importance of business process modeling and the role of errors in the business process management lifecycle.

Chapter 3: Event-driven Process Chains (EPC) This chapter gathers state-of-the-art work on EPCs. Building on the foundations of prior work, we establish a novel syntax definition and a novel semantics definition for EPCs. Our semantics are-based on transition relations that define both state changes and context changes. Furthermore, we present an algorithm to calculate the reachability graph of an EPC based on the transition relations and a respective implementation as a plug-in for ProM. The major motivations for this novel semantics are, firstly, semantic gaps and, secondly, non-intuitive behavior of existing formalizations.

Chapter 4: Verification of EPC Soundness This chapter presents an EPC-specific version of soundness as a criterion of correctness for EPCs. We propose two different approaches for the verification of soundness, one based on the reachability graph and another based on reduction rules. While the first approach explicitly considers all states and transitions that are represented by an EPC, there is a problem with state explosion, as the maximum number of states grows exponentially with the number of arcs. In order to avoid a performance problem, we introduce a set of reduction rules as second approach. This set extends prior work with new reductions for start and end components, delta components, prism components, and homogeneous EPCs. The second approach is tested by reducing the SAP Reference model. It shows that the reduction approach is *fast*, that it provides a *precise* result for almost all models, and that it finds *three times as many errors* as other approaches based on relaxed soundness.

Chapter 5: Metrics for Business Process Models This chapter discusses the suitability of business process model metrics to predict error probability from a theoretical point of view. Revisiting related research in the area of network analysis, software measurement, and metrics for business process models, we find that several aspects of process models are not yet combined in an overall measurement framework. Based on theoretical considerations, we present a set of 15 metrics related to size and 13 metrics that capture various aspects of the structure and the state space of the process model. For each of the metrics, we discuss their presumable connection with error probability and formulate respective hypotheses.

Chapter 6: Validation of Error Metrics In this chapter, we conduct several statistical analyses related to the connection between metrics and error probability. The results of the correlation analysis and the logistic regression model strongly confirm the hypothetical impact direction of the metrics. Furthermore, we derive a logistic regression function, based on a sample of about 2000 EPC business process models from practice, that correctly classifies 90% of the models from a second independent sample.

Chapter 7: Conclusions This chapter summarizes the findings and offers an outlook on future research. In particular, we discuss the implications of this thesis for guidelines and management for the business process modeling process, respective tool support, EPCs as a business process modeling language, and teaching of business process modeling.

a decision has to be made: whether to perform it in every process instance during run time (ON), or whether to exclude it permanently (OFF), i.e. it will not be executed in any process instance, or whether to defer this decision to run time (OPT), i.e. for each process instance, it has to be decided whether to execute the function or not. *Configurable connectors* subsume build-time connector types that are less or equally expressive. Hence, a configurable connector can only be configured to a connector type that restricts its behavior. A configurable OR-connector may be mapped to a regular OR-, XOR-, AND-connector, or to a single sequence of events and functions (indicated by SEQ_n for some process path starting with node n). A configurable AND-connector may only be mapped to a regular AND-connector. A configurable XOR-connector may be mapped to a regular XOR-connector or to a single sequence SEQ_n. Interdependencies between configurable EPC nodes can be specified via *configuration requirements*, i.e. logical expressions that define constraints for inter-related configuration nodes. *Configuration guidelines* formalize recommendations and best practices (also in the form of logical expressions) in order to support the configuration process semantically. Additional work formalizes C-EPC syntax [RA07], its mapping to EPCs [MRRA06], and its identification from existing systems [JVAR06].

3.4 EPC Semantics

In addition to related work on the syntax of EPCs, there are several contributions towards the formalization of EPC semantics. This section first illustrates the semantical problems related to the OR-join. Then it gives a historical overview of semantical definitions, and provides a formalization for EPCs, that is used in this thesis. Furthermore, we present an implementation of these semantics as a ProM plug-in that generates the reachability graph for a given EPC.

3.4.1 Informal Semantics as a Starting Point

Before discussing EPC formalization problems, we need to establish an informal understanding of state representation and state changes of an EPCs. Although we provide a formal definition not before Section 3.4.5, the informal declaration of state concepts helps to discuss formalization issues in this section. The *state*, or *marking*, of an EPC is defined by assigning a number of *tokens* (or process folders) to its arcs.[1] The formal semantics of an EPC define which state changes are possible for a given marking. These state changes are formalized by a *transition relation*. A node is called *enabled* if there are enough tokens on its incoming arcs that it can fire, i.e. a state change defined by a transition can be applied. This process is also called *firing*. A firing of a node n consumes tokens from its input arcs n_{in} and produces tokens at its output arcs n_{out}. The formalization of whether an OR-join is enabled is a non-trivial issue since not only the incoming arcs have to be considered. The sequence $\tau = n_1 n_2 ... n_m$ is called a firing sequence if it is possible to execute a sequence of steps, i.e. after firing n_1 it is possible to fire n_2, etc. Through a sequence of firings, the EPC moves from one *reachable* state to the next. The *reachability graph* of an EPC represents how states can be reached from other states. A marking that is not a final marking, but from which no other marking can be reached, is called a *deadlock*. The notion of an initial and a final marking will be formally defined in Section 3.4.5.

3.4.2 EPC Formalization Problems

We have briefly stated that the OR-join synchronizes all active incoming branches. This bears a non-trivial problem: if there is a token on one incoming arc, does the OR-join have to wait or not? Following the informal semantics of EPCs, it is only allowed to fire if it is not possible for a token to arrive on the other incoming arcs (see [NR02]). In the following subsection, we will show what the formal implications of these intended semantics are. Before that, we present some example EPCs, the discussion of which raises some questions that will not be answered immediately. Instead, we revisit them later on to illustrate the characteristics of different formalization approaches.

Figure 3.6(a) shows an EPC with an OR-join on a loop. There is a token on arc $a2$

[1] This state representation based on arcs reflects the formalization of *Kindler* [Kin03, Kin04, Kin06] and can be related to arcs between tasks in YAWL that are interpreted as implicit conditions [AH05]. Other approaches assign tokens to the nodes of an EPC, e.g., [Rum99]. Later, we will make a distinction between state and marking in our formalization of EPC operational semantics.

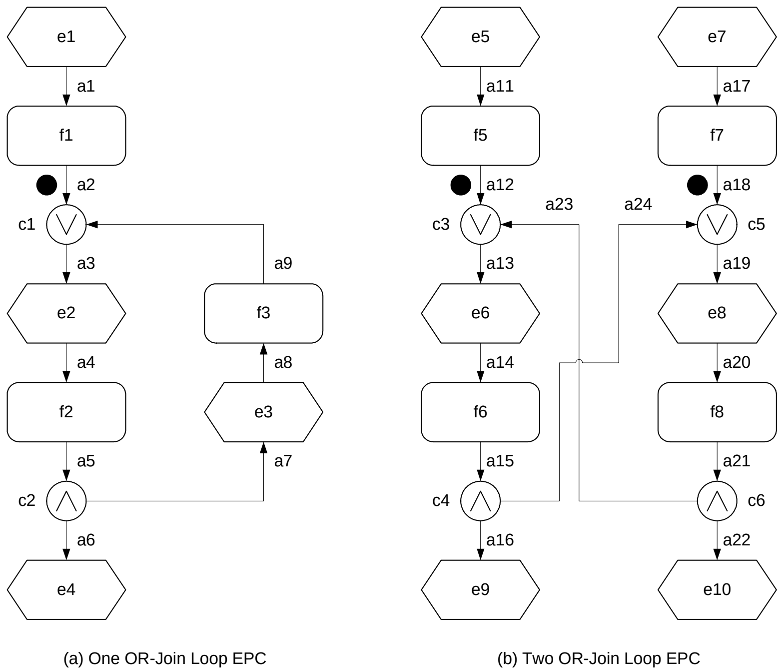


Figure 3.6: EPCs (a) with one OR-join and (b) with two OR-joins on the loop

from function $f1$ to the OR-join $c1$. The question is whether $c1$ can fire. If it could fire, then it would be possible for a token to arrive on arc $a9$ from $f3$ to the join. This would imply that it should wait and not fire. On the other hand, if it must wait, it is not possible that a token might arrive at $a9$. Figure 3.6(b) depicts an EPC with two OR-joins, $c3$ and $c5$, on a loop which are both enabled (cf. [ADK02]). Here, the question is whether both or none of them can fire. Since the situation is symmetrical, it seems unreasonable that only one of them should be allowed to fire.

The situation might be even more complicated, as Figure 3.7 illustrates (cf. [Kin03, Kin04, Kin06]). This EPC includes a loop with three OR-joins: $c1$, $c3$, and $c5$, all of which are enabled. Following the informal semantics, the first OR-join $c1$ is allowed to fire if it is not possible for a token to arrive on arc $a21$ from the AND-split $c6$. To put it differently, if $c1$ is allowed to fire, it is possible for a token to arrive on arc $a7$ that leads to the OR-join $c3$. Furthermore, the OR-join $c5$ could eventually fire. Finally, the first OR-join $c1$ would have to wait for that token before firing. To put it short, if $c1$ could fire, it would have to wait. One can show that this also holds the other way around: if it

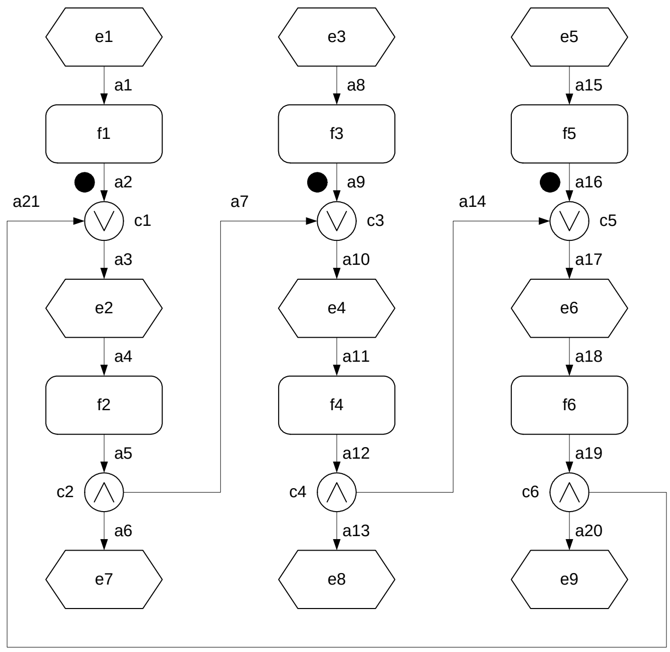


Figure 3.7: EPCs with three OR-joins on the loop

could not fire, it would not have to wait. Furthermore, this observation is also true for the two other OR-joins. In the subsequent section, we will discuss whether this problem can be resolved.

Refinement is another issue related to OR-joins. Figure 3.8 shows two versions of an EPC process model. In Figure 3.8(a) there is a token on $a7$. The subsequent OR-join $c2$ must wait for this token and synchronize it with the second token on $a5$ before firing. In Figure 3.8(b) the sequence $e3$-$a7$-$f3$ is refined with a block of two branches between an OR-split $c3a$ and an OR-join $c3b$. The OR-join $c2$ is enabled and should wait for the token on $a7f$. The question here is whether such a refinement might change the behavior of the OR-join $c1$. Figure 3.8 is just one simple example. The answer to this question may be less obvious if the refinement is introduced in a loop that already contains an OR-join. Figure 3.9 shows a respective case of an OR-join $c1$ on a loop that is refined with an OR-Block $c3a$-$c3b$. One would expect that the EPC of Figure 3.8(a) exhibits the same behavior as the one in (b). In the following section, we will discuss these questions from the perspective of different formalization approaches.

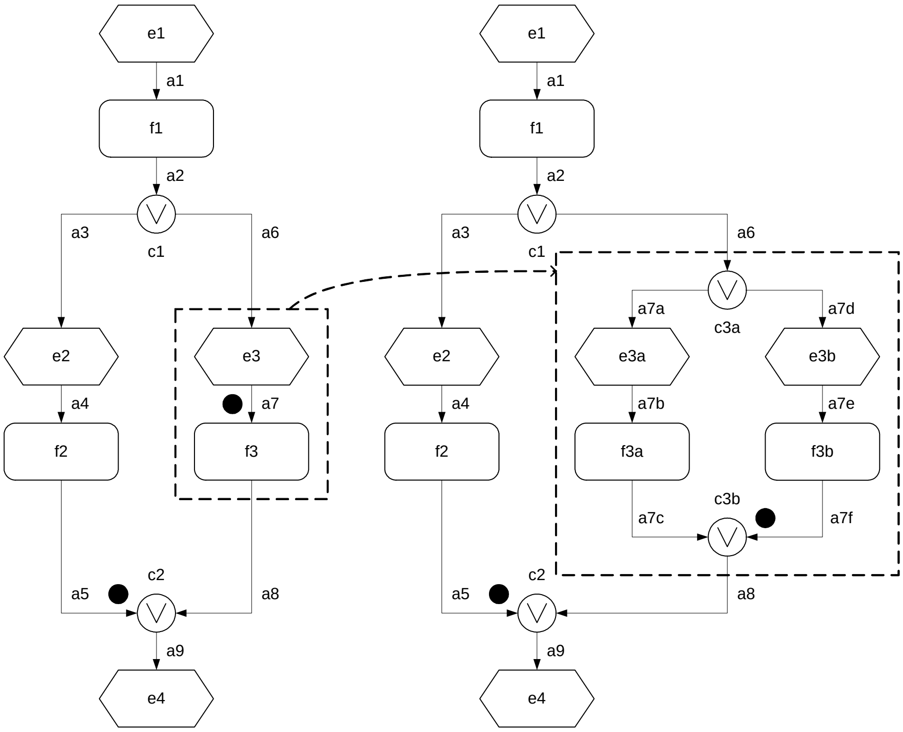
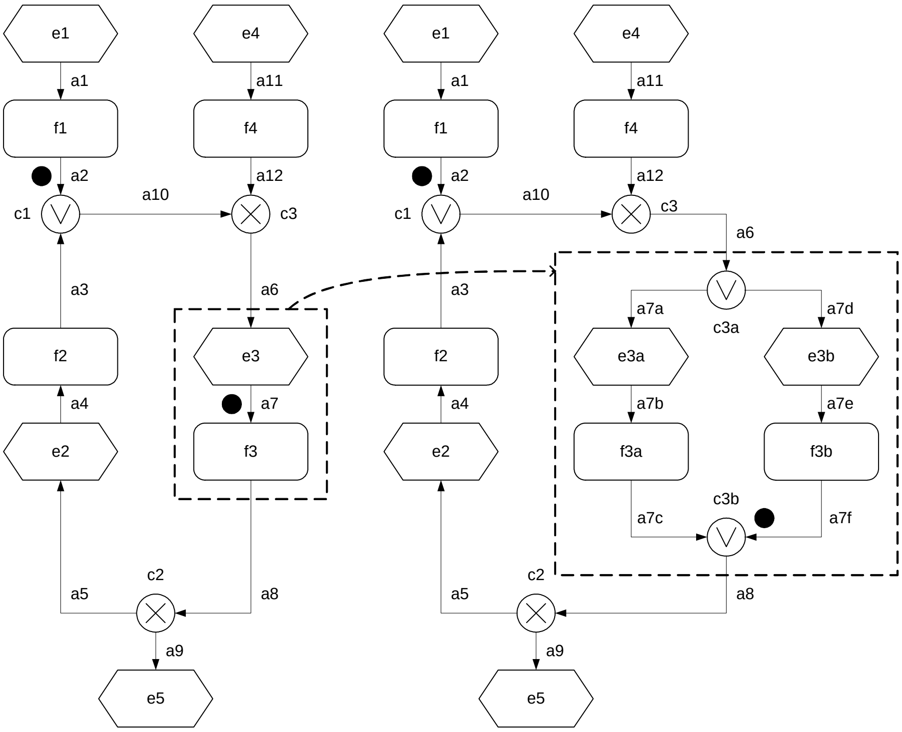


Figure 3.8: EPC refined with an OR-Block

3.4.3 Approaches to EPC Semantics Formalization

The transformation to Petri nets plays an important role in early formalizations of EPC semantics. In *Chen and Scheer* [CS94], the authors define a mapping to colored Petri nets and address the non-local synchronization behavior of OR-joins. This formalization builds on the assumption that an OR-split always matches a corresponding OR-join. The colored token that is propagated from the OR-split to the corresponding OR-join signals which combination of branches is enabled. Furthermore, the authors describe the state space of some example EPCs by giving reachability graphs. However, this first Petri net semantics for EPCs has mainly two weaknesses. Firstly, a formal algorithm to calculate the state space is missing. Secondly, the approach is restricted to EPCs with matching OR-split and OR-join pairs. Therefore, this approach does not provide semantics for the EPCs shown in figures 3.6 and 3.7. Even though the approach is not formalized in all its



(a) EPC with OR-join on loop

(b) Refinement of e3-f3 block

Figure 3.9: Cyclic EPC refined with an OR-Block

details, it should be able to handle the refined EPC of Figure 3.8(b) and the inner OR-join $c3b$ in Figure 3.8(b).

The transformation approach by *Langner, Schneider, and Wehler* [LSW97a, LSW97b, LSW98] maps EPCs to Boolean nets in order to define formal semantics. Boolean nets are a variant of colored Petri nets whose token colors are 0 (negative token) and 1 (positive token). Connectors propagate both negative and positive tokens according to their logical type. This mechanism is able to capture the non-local synchronization semantics of the OR-join similar to dead-path elimination in workflow systems (see [LA94, LR00]). The XOR-join only fires if there is one positive token on incoming branches and a negative token on all other incoming branches. Otherwise, it blocks. A drawback of this semantics definition is that the EPC syntax has to be restricted: arbitrary structures are not allowed. If there is a loop it must have an XOR-join as entry point and an XOR-split as exit point. This pair of connectors in a cyclic structure is mapped to one place in the resulting

Boolean net. As a consequence, this approach does not provide semantics for the EPCs in Figures 3.6 and 3.7. Still, it can cope with any pair of matching OR-split and OR-join. Accordingly, the Boolean nets define the expected semantics of the refined EPC of Figure 3.8(b) and of the inner OR-Block introduced as a refinement in Figure 3.8(b).

Van der Aalst [Aal99] presents a mapping approach to derive Petri nets from EPCs, but he does not give a mapping rule for OR-connectors because of the semantic problems illustrated in Section 3.4.2. The mapping provides clear semantics for XOR and AND-connectors as well as for the OR-split, but since the OR-join is not formalized, the approach does not provide semantics for the EPCs of Figures 3.6 to 3.9. *Dehnert* presents an extension of this approach by mapping the OR-join to a Petri net block [Deh02]. Since the resulting Petri net block may or may not necessarily synchronize multiple tokens at runtime (i.e., a non-deterministic choice), its state space is larger than the actual state space with synchronization. Based on the so-called relaxed soundness criterion, it is possible to cut away undesirable paths and, thus, check whether a join should be synchronized (cf. [DA04]).

In [Rit99, Rit00] *Rittgen* discusses the OR-join. He proposes to distinguish between three types of OR-joins on the syntactic level: every-time, first-come, and wait-for-all. The every-time OR-join basically reflects XOR-join behavior; the first-come OR-join passes the first incoming token and blocks afterwards; and the wait-for-all OR-join depends on a matching split similar to the approach of *Chen and Scheer*. This proposal could provide semantics for the example EPCs of Figures 3.6 to 3.9 in the following way. If we assume an every-time semantics, all OR-joins of the example EPCs could fire. While the loops would not block in this case, there would be no synchronization at all which contradicts the intended OR-join semantics. If the OR-joins behave according to the first-come semantics, all OR-joins could fire. Yet, there would also be no synchronization and the loops could be run only once. If the OR-joins had wait-for-all semantics, we would have the same problems as before with the loops. Altogether, the proposal by *Rittgen* does not provide a general solution to the formalization problem.

Building on prior work of *Rump* [ZR96, Rum99], *Nüttgens and Rump* [NR02] define a transition relation for EPCs that also addresses the non-local semantics of the OR-join, yet there is a problem: the transition relation for the OR-join refers to itself under negation. *Van der Aalst, Desel, and Kindler* [ADK02] show, that a fixed point for this transition relation does not always exist. They present an example to prove the opposite: an EPC with two OR-joins on a circle, which wait for each other as depicted in Figure 3.6(b). This vicious circle is the starting point for the work of *Kindler* towards a sound mathematical framework for the definition of non-local semantics for EPCs. In a series of papers [Kin03, Kin04, Kin06], *Kindler* elaborates on this problem in detail. The technical problem is that for the OR-join the transition relation R depends upon itself in negation. Instead of defining one transition relation, he considers a pair of transition relations (P, Q) on the state space Σ of an EPC and a monotonously decreasing function $R : 2^{\Sigma \times N \times \Sigma} \rightarrow 2^{\Sigma \times N \times \Sigma}$. Then, a function $\varphi((P, Q)) = (R(Q), R(P))$ has a least fixed point and a greatest fixed point. P is called pessimistic transition relation and Q optimistic transition relation. An EPC is called *clean*, if $P = Q$. For most EPCs, this is the case. Some EPCs, such as the vicious circle EPC, are *unclean* since the pessimistic and the optimistic semantics do not coincide. Moreover, *Cuntz* provides an example of a clean EPC, which becomes unclean by refining it with another clean EPC [Cun04, p.45]. *Kindler* even shows that there are acyclic EPCs that are unclean (see [Kin06, p.38]). Furthermore, *Cuntz and Kindler* present optimizations for an efficient calculation of the state space of an EPC, and a respective prototype implementation called EPC Tools [CK04, CK05]. EPC Tools also offers a precise answer to the questions regarding the behavior of the example EPCs in Figures 3.6 to 3.9.

- Figure 3.6(a): For the EPC with one OR-join on a loop, there is a fixed point and the connector is allowed to fire.
- Figure 3.6(b): The EPC with two OR-joins on a loop is unclean. Therefore, it is not clear whether the optimistic or the pessimistic semantics should be considered.
- Figure 3.7: The EPC with three OR-joins is also unclean, i.e., the pessimistic deviates from the optimistic semantics.
- Figure 3.8(a): The OR-join $c2$ must wait for the second token on $a7$.
- Figure 3.8(b): The OR-join $c2$ must wait for the second token on $a7f$.
- Figure 3.9(a): The OR-join $c1$ must wait for the second token on $a7$.
- Figure 3.9(b): The OR-join $c1$ is allowed to fire, the second OR-join $c2$ in the OR-block must wait.

Even though the approach by *Kindler* provides semantics for a large subclass of EPCs, i.e. clean EPCs, there are some cases like the EPCs of Figure 3.6(b) and 3.7 that do not have semantics. The theorem by *Kindler* proves that it is not possible to calculate these EPCs semantics as long as the transition relation is defined with a self-reference under negation. Furthermore, such a semantics definition may imply some unexpected results, e.g. the EPC of Figure 3.9(a) behaves differently than its refinement in Figure 3.9(b).

While it is argued that unclean EPCs only have theoretical relevance, there actually are unclean EPCs in practice. Figure 3.10 shows the Test Equipment Management EPC from the Quality Management branch of the SAP Reference Model (cf. [KT98]). The marking of this EPC in the figure can be produced by firing the OR-split on the right-hand side of the EPC. Both XOR-joins are on a loop resulting in an unclean marking. This illustrates the need in theory *and* practice to also provide semantics for EPCs that are unclean, according to *Kindler's* definition [Kin06].

Van Hee, Oanea, and Sidorova discuss a formalization of extended EPCs as they are implemented in the simulation tool of the ARIS Toolset (see [IDS03a]) based on a transition system [HOS05]. This transition system is mapped to colored Petri nets in order to do verification with CPN Tools (see [JKW07]). The considered EPC extension includes data attributes, time, and probabilities which are used for the simulation in ARIS. The essential idea of this formalization and the ARIS implementation is that process folders can have timers, and that these timers are used at an OR-join for synchronization purposes.[2] If a folder arrives at an OR-join it has to wait until its timer expires. Since the timers are only reduced if there are no folders to propagate, the OR-join can synchronize multiple incoming folders. A problem of this approach is that once the timer of a folder is expired, there is no way to synchronize it once it has passed the OR-join. If there are several OR-joins used in sequence, only the first one will be synchronized. Therefore, this formalization – though being elaborate – provides only a partial solution to the formalization of the OR-join.

Van der Aalst and Ter Hofstede defined a workflow language called YAWL [AH05] which also offers an OR-join with non-local semantics. As *Mendling, Moser, and Neu-*

[2]Note that this general approach can be parameterized in ARIS with respect to sychronization and waiting semantics (see [HOS05, p.194]).

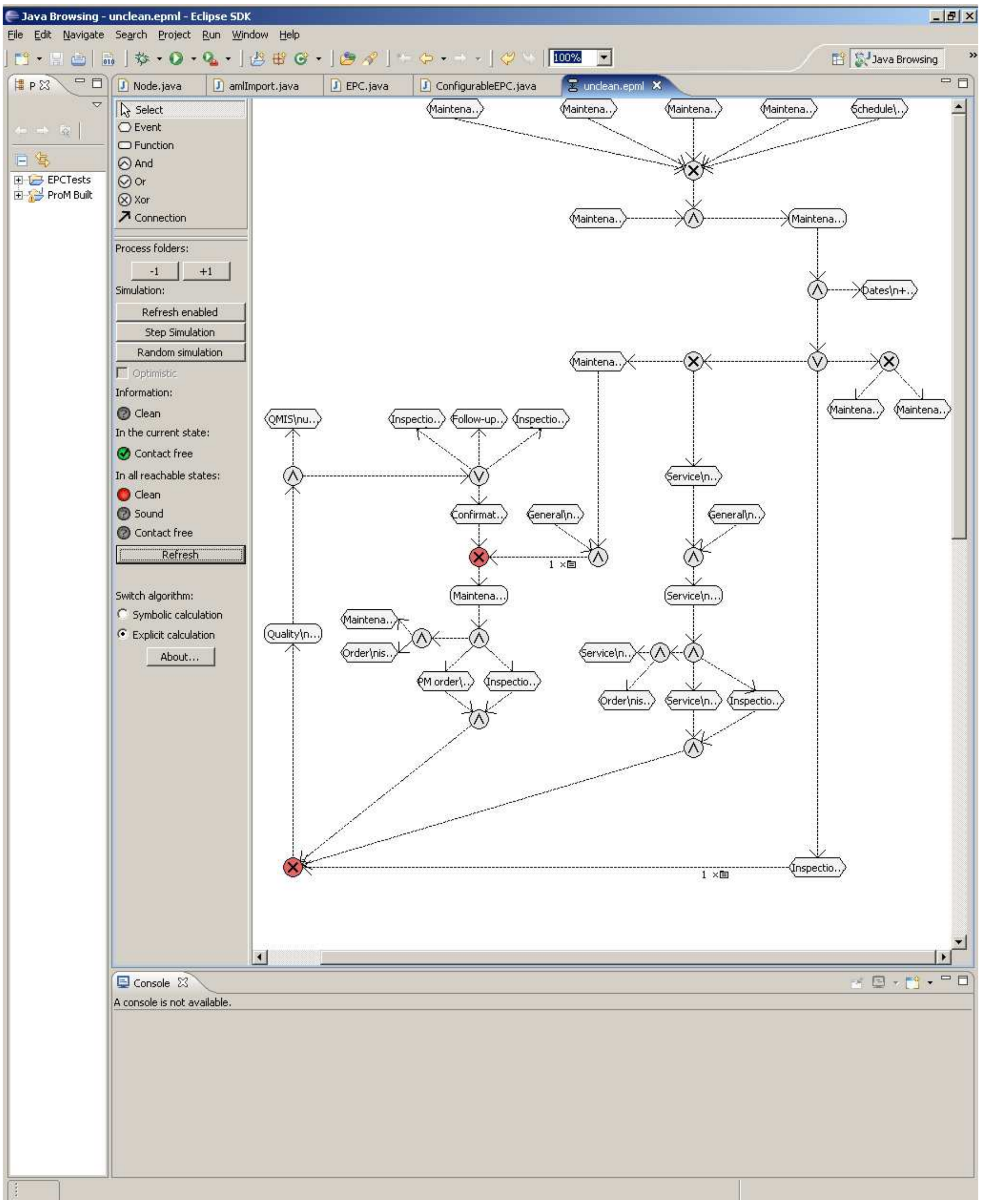


Figure 3.10: Test Equipment Management EPC from the Quality Management branch of the SAP Reference Model

mann propose a transformation semantics for EPCs based on YAWL [MMN06a], we will discuss how the OR-join behavior is formalized in YAWL. In [AH05], the authors propose a definition of the transition relation $R(P)$ with a reference to a second transition relation P that ignores all OR-joins. A similar semantics that is calculated on history-logs of the process is proposed by *Van Hee, Oanea, Serebrenik, Sidorova, and Voorhoeve* in [HOS$^+$06]. The consequence of this definition can be illustrated using the example EPCs.

- Figure 3.6(a): The single OR-join on the loop can fire.
- Figure 3.6(b): The two OR-joins on the loop can fire.
- Figure 3.7: The three OR-joins on the loop can fire.
- Figure 3.8(a): The OR-join $c2$ must wait for the second token between e3 and f3.
- Figure 3.8(b): Both OR-joins can fire.
- Figure 3.9(a): The OR-join $c1$ must wait for the second token between e3 and f3.
- Figure 3.9(b): Both OR-joins can fire.

Kindler criticizes that each choice for defining P "appears to be arbitrary or ad hoc in some way" [Kin06] and uses the pair (P, Q) instead. The example EPCs illustrate that the original YAWL semantics provide for a limited degree of synchronization. Consider, for example, the vicious circle EPC with three OR-joins: all are allowed to fire, but if one does, the subsequent OR-join has to wait. Furthermore, the refined EPCs exhibit different behavior from their unrefined counterparts since OR-joins are ignored, i.e. they are considered unable to fire.

Wynn, Edmond, Van der Aalst, and Ter Hofstede illustrate that the OR-join semantics in YAWL exhibit some non-intuitive behavior when OR-joins depend upon each other [WEAH05]. Therefore, they present a novel approach based on a mapping to Reset nets. Whether or not an OR-join can fire (i.e. $R(P)$), is determined depending on (a) a corresponding Reset net (i.e. P) that treats all OR-joins as XOR-joins[3], and (b) a predicate

[3] In fact, [WEAH05] proposes two alternative treatments for the "other OR-joins" when evaluating an OR-join: treat them either as XOR-joins (optimistic) or as AND-joins (pessimistic). However, the authors select the optimistic variant because the XOR-join treatment of other OR-joins more closely match the informal semantics of the OR-join.

called *superM* that prevents firing if an OR-join is on a directed path from another enabled OR-join. In particular, the Reset net is evaluated using backward search techniques that grant coverability to be decidable (see [LL00, FS01]). A respective verification approach for YAWL nets is presented in [WAHE06]. Using these semantics, the example EPCs behave as follows:

- Figure 3.6(a): The single OR-join on the loop can fire since *superM* evaluates to false, and hence no more tokens can arrive at c_1.
- Figure 3.6(b): The two OR-joins are not enabled since *superM* evaluates to true, because if the respectively other OR-join is replaced by an XOR-join, an additional token may arrive.
- Figure 3.7: The three OR-joins are not enabled, because if one OR-join assumes the other two to be XOR-joins, then this OR-join has to wait.
- Figure 3.8(a): The OR-join $c2$ must wait for the second token on *a7*.
- Figure 3.8(b): The OR-join $c2$ must wait for the second token on *a7f*.
- Figure 3.9(a): The OR-join *c1* must wait for the token on *a7*.
- Figure 3.9(b): The OR-join *c1* must wait because if $c3b$ is assumed to be an XOR-join a token may arrive via $a3$. The OR-join *c3b* must also wait, because if $c1$ is an XOR-join, another token may move to $a7c$. Therefore, there is a deadlock.

The novel approach based on Reset nets provides interesting semantics, but in some cases also leads to deadlocks.

Table 3.3: Overview of EPC semantics and their limitations

OR-join semantics	Limitations
[CS94]	OR-join must match OR-split
[LSW98]	Joins as loop entry undefined
[Rit99] every-time	missing synchronization
[Rit99] first-come	OR-join can block
[Rit99] wait-for-all	OR-join as loop entry undefined
[Kin06]	EPC can be unclean
[HOS05]	folders with expired timers do not synchronize
[AH05]	limited synchronization
[WAHE06]	OR-join may block

Table 3.3 summarizes existing work on the formalization of the OR-join. Several early approaches define syntactical restrictions, such as OR-splits, to match corresponding OR-joins or models to be acyclic (see [CS94, LSW98, Rit99]). Newer approaches impose little or even no restrictions (see [Kin06, AH05, WAHE06]), but exhibit unexpected behavior for OR-block refinements on loops with further OR-joins on it. The solution based on Reset nets seems to be most promising from the intuition of its behavior. Yet, it requires extensive calculation effort since it depends upon backward search to decide coverability (Note that reachability is undecidable for reset nets illustrating the computational complexity of the OR-join in the presence of advanced routing constructs). In the following subsection, we propose a novel approach that overcomes some of the refinement problems of the Reset nets semantics and that provides a more intuitive solution since all OR-join decisions can be made with local knowledge.

3.4.4 A Novel Approach towards EPC Semantics

In this subsection, we introduce a novel concept for EPC semantics.[4] The formalization of this concept follows in the subsequent section. The principal idea of these semantics borrows some concepts from *Langner, Schneider, and Wehler* [LSW98] and adapts the idea of Boolean nets with true and false tokens in an appropriate manner. Furthermore, we utilize similar notations as *Kindler* [Kin06], modifying them where needed. The transition relations that we will formalize afterwards depend on the state and the context of an EPC. The *state* of an EPC is basically an assignment of positive and negative tokens to the arcs. Positive tokens signal which functions have to be carried out in the process, negative tokens indicate which functions are to be ignored for the moment. The transition rules of AND-connector and OR-connectors are adopted from the Boolean nets formalization which facilitates synchronization of OR-joins in structured blocks. In order to allow for a more flexible utilization of XOR-connectors in a cyclic structure, we modify and extend the approach of Boolean nets in three ways:

[4]An earlier version of these semantics is described in [MA06]. Essentially, this version is different in two ways: (1) Dead context is propagated already if only one input is dead. Without that, XOR-loops could not be marked dead. (2) We introduce a concept to clean up negative tokens that could not be forwarded to an OR-join (see *negative upper corona* in phase 4 for positive token propagation).

1. XOR-splits produce one positive token on one of their their output arcs, but no negative tokens. XOR-joins fire each time there is a positive token on an incoming arc. This mechanism provides the expected behavior in both structured XOR-loops and structured XOR-blocks where an XOR-split matches an XOR-join.
2. In order to signal OR-joins that it is not possible to have a positive token on an incoming branch, we define the *context* of an EPC. The context assigns a status of *wait* or *dead* to each arc of an EPC. A wait context indicates that it is still possible that a positive token might arrive; a dead context status means that either a negative token will arrive next, or no positive token can arrive anymore. For example, XOR-splits produce a dead context on those output branches that are not taken, and a wait context on the output branch that receives a positive token. A dead context at an input arc is then used by an OR-join to determine whether it has to synchronize with further positive tokens or not. Since dead and wait context might be conflicting and, thus, have to alternate, both context and state is propagated in separate phases to guarantee termination.
3. The propagation of context status and state tokens is arranged in a four phase cycle: (a) dead context, (b) wait context, (c) negative token, and (d) positive token propagation.

   (a) In this phase, all *dead context* information is propagated in the EPC until no new dead context can be derived.

   (b) Then, all *wait context* information is propagated until no new wait context can be derived. It is necessary to have two phases (i.e., first the dead context propagation and then the wait context propagation) in order to avoid infinite cycles of context changes (details below).

   (c) After that, all *negative tokens* are propagated until no negative token can be propagated anymore. This phase cannot run into an endless loop (details below).

   (d) Finally, one of the enabled nodes is selected and propagates *positive tokens* leading to a new iteration of the four phase cycle.

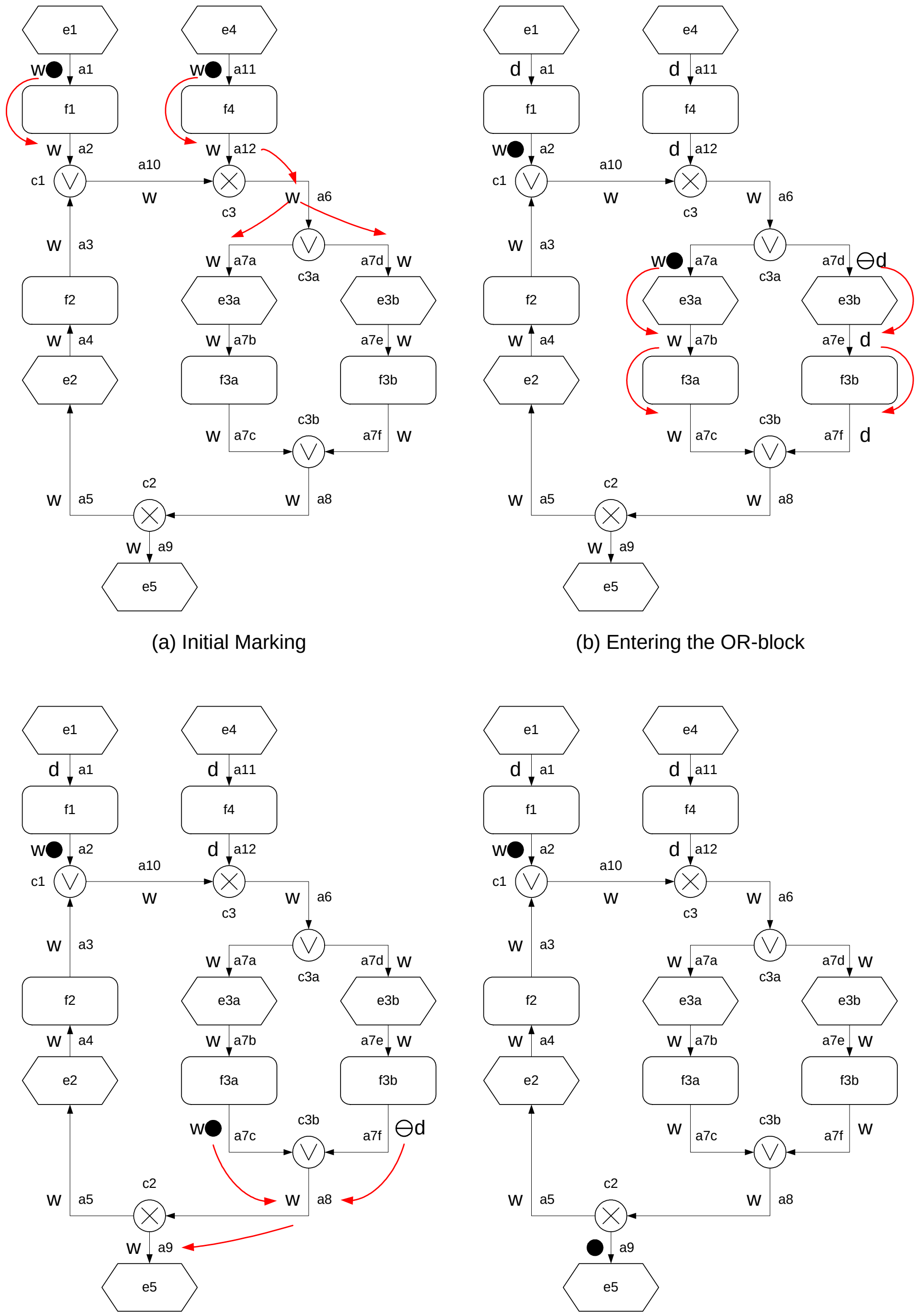
In the following part, we first give an example to illustrate the behavior of the EPC semantics before defining state, context, and each transition phase in detail.

Revisiting the cyclic EPC refined with an OR-block

Figure 3.11 revisits the example of the cyclic EPC refined with an OR-block that we introduced as Figure 3.9 in Section 3.4.2.

In Figure 3.11(a), an initial marking with two positive tokens on $a1$ and $a11$ is given. These positive tokens induce a wait context on all arcs, which implies that all of them might potentially receive a positive token at some point in time. The context status is indicated by a letter next to the arc: w for wait and d for dead. Subsequently, the positive tokens can be propagated to the arcs $a2$ and $a12$, respectively and the context of $a1$ and $a11$ changes to dead. In this situation, the OR-join $c1$ is not allowed to fire due to the wait context on arc $a3$, but has to synchronize with positive and negative tokens that might arrive there. The XOR-join is allowed to fire without considering the second arc $a10$. In (b) the OR-split $c3a$ has fired (following the execution of c3) and produces a positive token on $a7a$ and a negative token on $a7d$. Accordingly, the context of $a7d$ is changed to dead. This dead context is propagated down to arc $a7f$. The rest of the context remains unchanged. The state shown in (b) is followed by (c) where the positive and the negative tokens are synchronized at the connector $c3b$ and one positive token is produced on the output arc $a8$. Please note that the OR-join $c3b$ does not synchronize with the other OR-join $c1$ that is also on the loop. In the *Kindler* and the Reset nets semantics, $c3b$ would have to wait for the token from $a2$. Here, the wait context propagation is blocked by the negative token. In (d), the XOR-split $c2$ produces a positive token on $a9$ and a dead context on $a5$. This dead context is propagated via $a3$ to the rest of the loop in the dead context propagation phase. In the wait context propagation phase, the dead context of the loop is reset to wait, which is propagated from $c1$. As a consequence, the OR-join $c1$ is not enabled.

This example allows us to make two observations. Firstly, the context propagation blocks OR-joins that are entry points to a loop in a wait position since the self-reference is not resolved. Secondly, the XOR-split produces a dead context, but not a negative



(a) Initial Marking (b) Entering the OR-block

(c) Synchronizing (d) Exiting the loop

Figure 3.11: Example of EPC marking propagation

token. The disadvantage of producing negative tokens would be that the EPC is flooded with negative tokens if an XOR-split was used as an exit of a loop. These tokens would give downstream joins the wrong information about the state of the loop, since it would still be live. An OR-join could then synchronize with a negative token while a positive token is still in the loop. In contrast to that, the XOR-split as a loop exit produces a dead context. Since there is a positive token in the loop, it overwrites the dead context at the exit in the wait context propagation phase. Downstream OR-joins then have the correct information that there are still tokens to wait for.

Definition of State, Context, and Marking

We define both state and context as an assignment to the arcs. The term *marking* refers to state and context together. The EPC transition relations defines which state and/or context changes are allowed for a given marking in a given phase.

Definition 3.12 (State and Context)**.** Let $EPC = (E, F, C, l, A)$ be a standard EPC. Then, a mapping $\sigma : A \rightarrow \{-1, 0, +1\}$ is called a state of an EPC. The positive token captures the state as it is observed from outside the process. It is represented by a black filled circle. The negative token depicted by a white open circle with a minus on it has a similar semantics as the negative token in the Boolean nets formalization. Arcs with no state tokens on them do not depict a circle. Furthermore, a mapping $\kappa : A \rightarrow \{wait, dead\}$ is called a context of an EPC. A wait context is represented by a w and a dead context by a d next to the arc.

In contrast to Petri nets, we distinguish the terms *marking* and *state*: the term marking refers to state σ and context κ collectively.

Definition 3.13 (Marking of an EPC)**.** Let $EPC = (E, F, C, l, A)$ be a standard EPC. Then, a mapping $m : A \rightarrow (\{-1, 0, +1\} \times \{wait, dead\})$ is called a marking. The set of all markings M_{EPC} of an EPC is called marking space with $M_{EPC} = A \times \{-1, 0, +1\} \times \{wait, dead\}$. The projection of a given marking m to a subset of arcs $S \subseteq A$ is referred to as m_S. The marking m_a of an arc a can be written as $m_a = (\kappa(a) + \sigma(a)) \cdot a$, e.g. $(w + 1)a$ for an arc with a wait context and a positive token. If we refer to the κ- or the σ-part of m, we write κ_m and σ_m, respectively, i.e. $m(a) = (\sigma_m(a), \kappa_m(a))$.

Initial and Final Marking

The initial marking is the starting point for applying an iteration of the four-phase cycle. In [Rum99], the initial marking of an EPC is specified as an assignment of tokens to one, some, or all start events. While such a definition contains enough information for verification purposes, for example by the bundling of start and end events with OR-connectors as proposed in [MMN06a], it does not provide executable semantics according to the original definition of EPCs. As pointed out in [Rit99], it is not possible to equate the triggering of a single start event with the instantiation of a new process. This is because EPC start events do not only capture the creation of a process instance, but also external events that influence the execution of a running EPC (cf. [CS94]). This observation suggests an interactive validation approach as presented by [DAV05], where the user makes explicit assumptions about potential combinations of start events. In our approach, we assume that in the initial marking, all start arcs $a_s \in A_s$ have either a positive or a negative token with the matching context[5]. A respective formalization of initial and final marking is given later in Definitions 3.14 and 3.15. In the following sections, we describe the transition relations of each node $n \in E \cup F \cup C$ in the phases of dead context, wait context, negative and positive token propagation.

Phase 1: Dead Context Propagation

The transition relation for dead context propagation defines rules for deriving a dead context if one input arc of a node has a dead context status. Note that this rule might result in arcs having a dead context that could still receive a positive token. Those arcs are reset to a wait context in the subsequent phase of wait context propagation (Phase 2).

Figure 3.12 gives an illustration of the transition relation. *Please note that the figure does not depict the fact that the the rules for dead context propagation can only be applied if the respective output arc does not hold a positive or a negative token.* Concrete tokens override context information, for isntance, an arc with a positive token will always

[5] The context of non-start arcs is derived when the four propagation phases are entered the first time. We choose to initialize all non-start arcs with a wait context (cf. Figure 3.11). Note that this context might be changed in the dead context propagation phase before any token is moved.

have a wait context. Rules (a) and (b) indicate that if an input arc of a function or an event is dead, then also the output arc has to have a dead context status. Rule (c) represents that each split-connector propagates a dead context to its output arcs. These transition relations formalize the observation that if an input arc cannot receive a token anymore, this also holds true for its output arcs (unless they already hold positive or negative tokens). The join-connectors require only one dead context status at their input arcs for reproducing it at their output arc, see (d). It is important to note that a dead context is propagated until there is an end arc or an arc that carries a token.

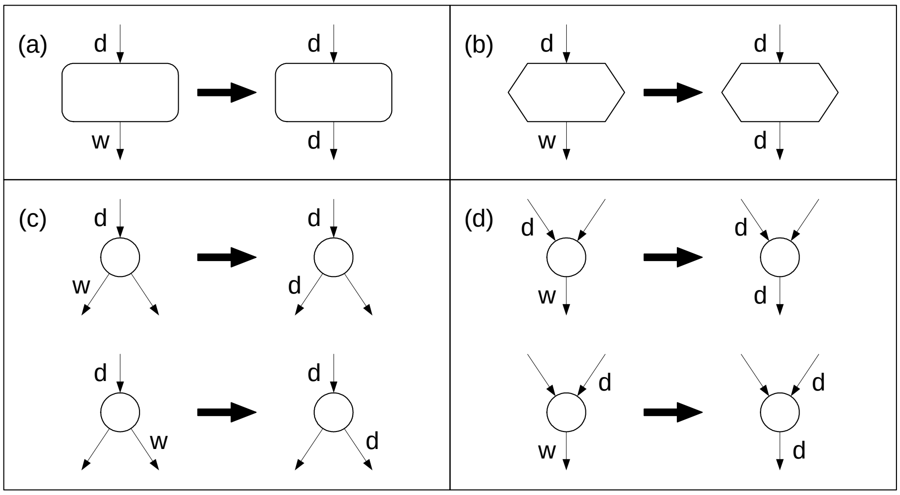


Figure 3.12: Transition relation for dead context propagation

Phase 2: Wait Context Propagation

The transition relation for wait context propagation defines rules for deriving a wait context if one or more input arcs of a node have a wait context status. Figure 3.13 gives an illustration of the transition relation. *All transitions can only be applied if the respective output arc does not hold a positive or a negative token.* Concrete tokens override context information, i.e. an arc with a positive token will always have a wait context. Rules (a) and (b) show that if an input arc of a function or an event has a wait context, then the output arc also has to have a wait context status. Rule (c) represents that each split-connector propagates a wait context to its output arcs. The AND-join requires all inputs to have a wait context status in order to reproduce it at its output arc, see (d). XOR- and OR-joins propagate a wait context if one of their input arcs has a wait context, see (e) and (f). Similar to the dead context propagation, the wait context is propagated until an end node is received or until an arc holds a token. Furthermore, the wait context is propagated by an AND-join where all of the inputs have a wait context.

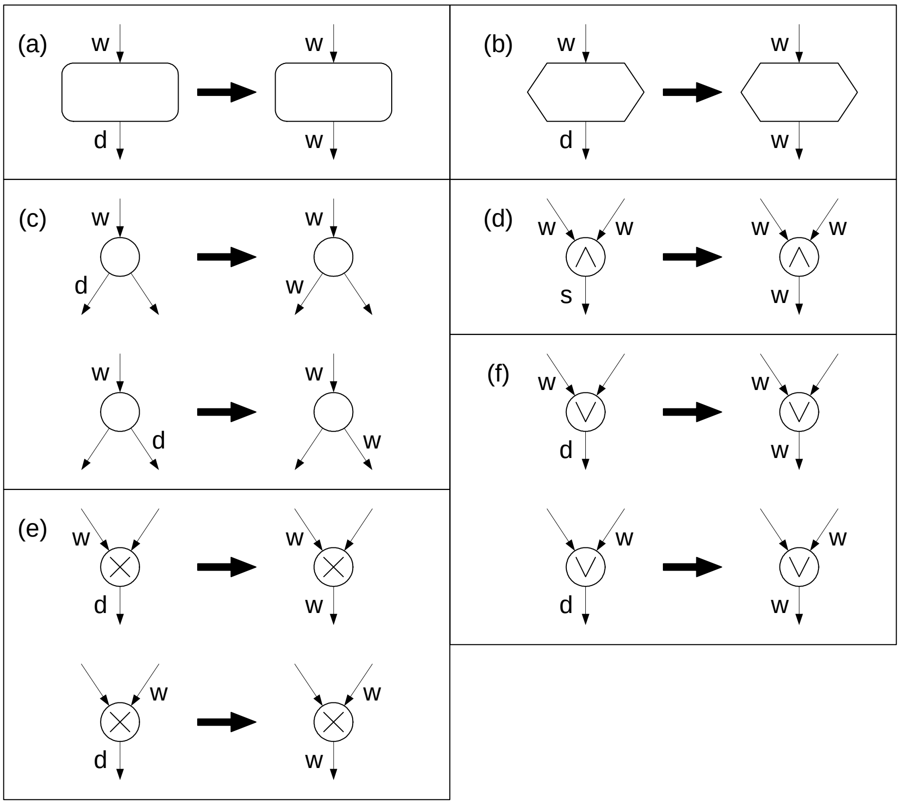


Figure 3.13: Transition relation for wait context propagation

Observations on Context Propagation

The transition relations of context propagation permit the following observations:

- *Context changes terminate:* It is intuitive that context propagation cannot run in an infinite loop. It is easy to verify that the first two phases stop. The propagation of dead context stops because the number of arcs is finite, i.e., the number of arcs is an upper bound for the number of times the rules in Figure 3.12 can be applied. A similar argument applies to the propagation of the wait context. As a consequence, the context change phase will always terminate and enable the consideration of new state changes in the subsequent phase.

- *State tokens block context propagation:* The transition relations for context propagation require that the output arcs to be changed do not hold any state token, i.e., arcs with a positive token always have a wait context, and arcs with a negative token always have a dead context.
- *Context propagating elements:* Functions, events, and split nodes reproduce the context that they receive at their input arcs.
- *OR- and XOR-joins:* Both these connectors reproduce a dead and also a wait context if at least one of the input arcs has the respective context.
- *AND-joins:* AND-joins produce wait context status only if all inputs are wait. Otherwise, the output context remains in a dead context.

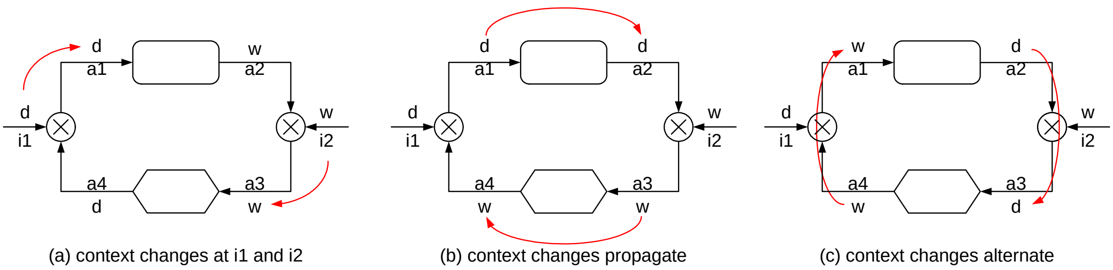


Figure 3.14: Situation of unstable context changes without two phases

Figure 3.14 illustrates the need to perform context propagation in two separate phases as opposed to together in one phase. If there are context changes (a) at $i1$ and $i2$, the current context enables the firing of the transition rules for both connectors producing a $dead$ context status in $a1$ and a $wait$ context status in $a3$. This leads to a new context in (b) with an additional $dead$ context status in $a2$ and a new $wait$ context status in $a4$. Since both arcs from outside the loop to the connectors are marked in such a way that incoming context changes on the other arc is simply propagated, there is a new context in (c) with a $wait$ status in $a1$ and a $dead$ context status in $a3$. Note that this new context can be propagated, and this way the initial situation is reproduced. This can be repeated again and again. Without a sequence of two phases, the transitions could continue infinitely and the result would be undefined.

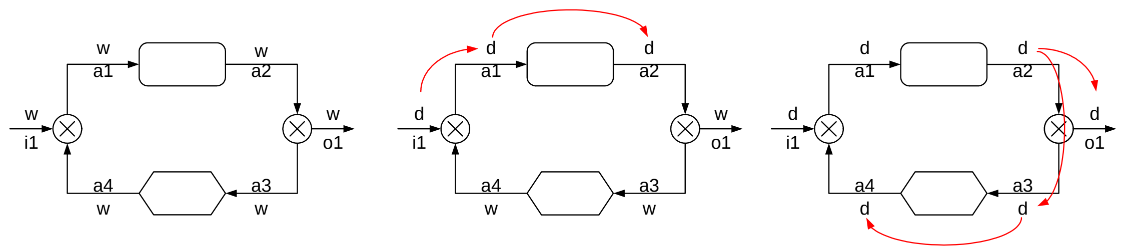


Figure 3.15: Propagating dead context in a loop

The precedence of the two phases can also be motivated using an example EPC containing a loop. The propagation of dead context with only one dead input is needed to accurately mark loops as dead. Figure 3.15 shows the picture of a simple loop with one XOR-join as entrance and one XOR-split as exit. Initially, the loop might be in a wait context (a). If the path to the loop becomes dead, this context is propagated into the loop (b) and to its output (c). If not all join-connectors would propagate dead context with one dead input, the loop could never become dead. But since this often results in too many dead arcs, the wait context propagation must be performed afterwards. It guarantees that arcs that can still be receive a positive token get a wait context.

Phase 3: Negative Token Propagation

Negative tokens can result from branches that are not executed after OR-joins or start events. The transition relation for negative token propagation includes four firing rules that consume and produce negative tokens. Furthermore, the output arcs are set to a dead context. Figure 3.16 gives an illustration of the transition relation. *All transitions can only be applied if all input arcs hold negative tokens and if there is no positive token on the output arc.* In the following section, we will show that this phase terminates.

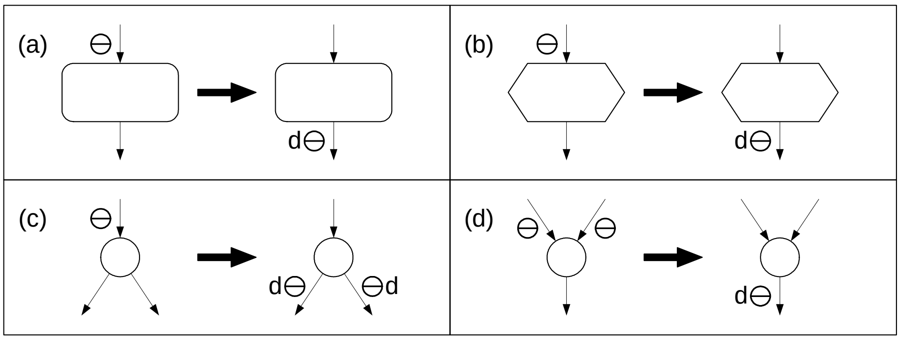


Figure 3.16: Transition Relation for Negative Token Propagation

Phase 4: Positive Token Propagation

The transition relation for positive token propagation specifies firing rules that consume negative and positive tokens from the input arcs of a node to produce positive tokens on its output arcs. Figure 3.17 gives a respective illustration. Rules (a) and (b) show that functions and events consume positive tokens from the input arc and propagate them to the output arc. Furthermore, and this holds true for all rules, consuming a positive token from an arc implies setting this arc to a dead context status. Rules (c) and (d) illustrate that AND-splits consume one positive token and produce one on each output arc, while AND-joins synchronize positive tokens on all input arcs to produce one on the output arc. Rule (e) depicts the fact that XOR-splits forward positive tokens to one of their output arcs. In contrast to the Boolean net formalization, they do not produce negative tokens, but a dead context on the output arcs which do not receive the token. Correspondingly, XOR-joins (f) propagate each incoming positive token to the output arc, no matter what the context or the state of the other input arcs is. If there are negative tokens on the incoming arcs, they are consumed. The OR-split (g) produces positive tokens on those output arcs that have to be executed and negative tokens on those that are ignored. Note that the OR-join is the only construct that may introduce negative tokens (apart from start events that hold a negative token in the initial marking). Rule (h) shows that on OR-join can only fire either if it has full information about the state of its input arcs (i.e., each input has a positive or a negative token) or all arcs that do not hold a token are in a dead context. Finally, in all rules, each output arc that receives a negative token is set to a dead context and each that gets a positive token is set to a wait context.

The last two firing rules of the OR-join in Figure 3.17(h) deserve some further comments. Beyond the removal of all positive and negative tokens on the input arcs, also those negative tokens on the *negative upper corona* of the OR-join are removed. The motivation for this concept is that loops can propagate dead context, but negative tokens get stuck at the entry join of a loop. After the loop, a dead context can make the firing condition of an OR-join become true, while negative tokens that were generated for synchronization purposes still reside before the loop. Not removing such negative tokens with the firing of an OR-join might cause non-intuitive behavior. Therefore, in addition to the positive and negative tokens on the input arcs of the OR-join, also those negative tokens with a path leading to the OR-join via arcs that all have a dead context, i.e. on the negative corona, are also removed.

Figure 3.18 gives the example of a structured EPC with an outer XOR-loop between $c1$ and $c6$ and an inner XOR-loop between $c3$ and $c4$. The inner loop is also nested in an OR-block between $c2$ and $c5$. The current marking is produced by firing the OR-split with a negative token to the left and a positive token to the right, and then propagating the positive token via $f2$. Now, the OR-join $c5$ is enabled with a dead context on one of the input arcs. Moreover, there is a negative token before the inner XOR-loop which cannot be propagated. If the OR-join would now simply fire and navigate via $e2$ back to $c2$ the EPC would be in a deadlock since the firing rules for tokens require the output arcs to be empty. Therefore, the negative token before $c3$ has to be removed when firing the OR-join $c5$. Accordingly, if an OR-join fires, it has to remove all negative tokens on its so-called negative upper corona, i.e. the arcs carrying a negative token that have a path to the OR-join on which each arc has a dead context and no token on it.

Observations on State Propagation

The transition relations of state propagation permit the observation that the EPC semantics are *safe*, i.e. it is not possible to have more than one token on an arc. Firstly, this property is enforced by the definition of state since it is a mapping from the arcs to the set of -1,0, and +1. Furthermore, the state propagation rules guarantee safeness, too,

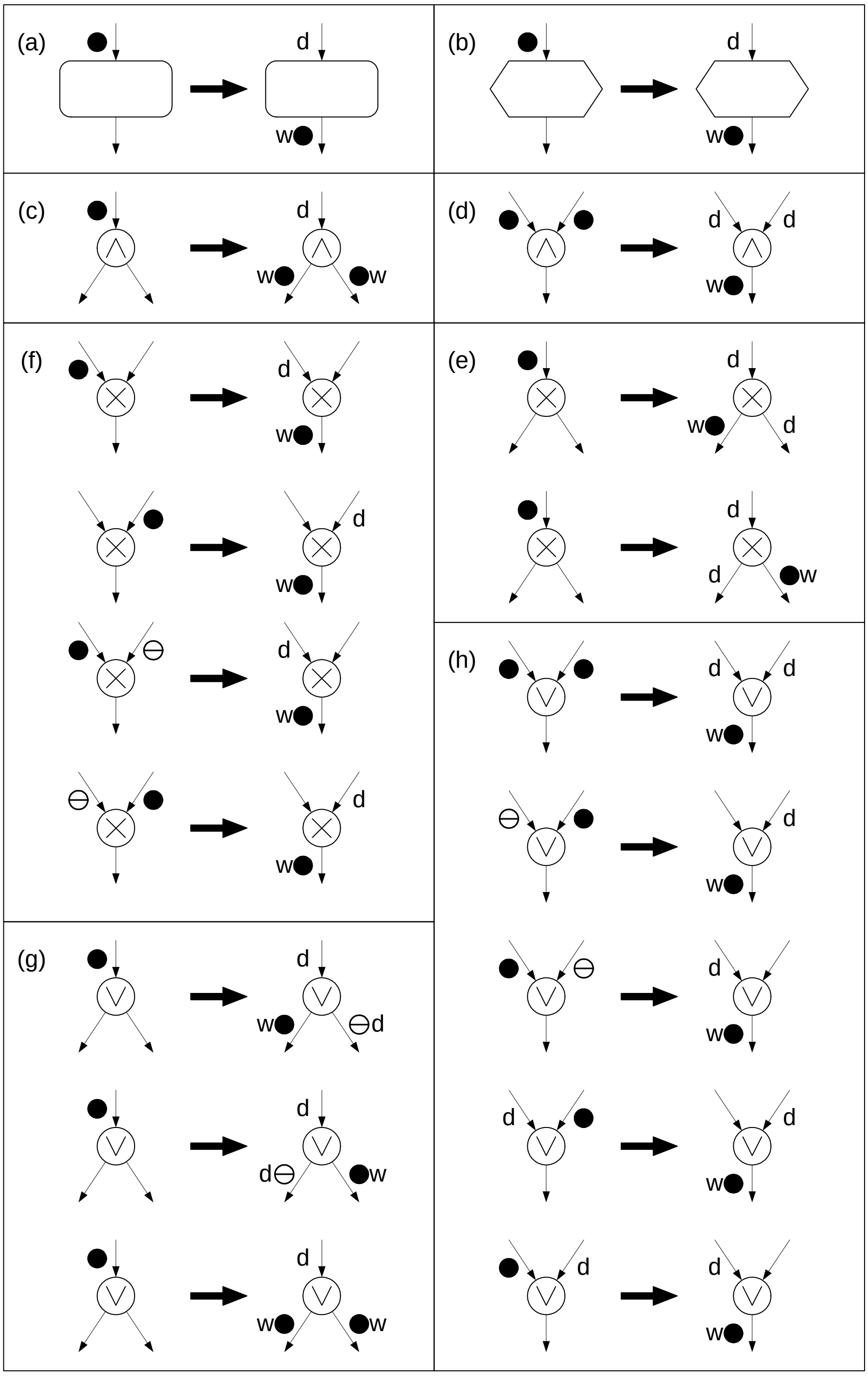


Figure 3.17: Transition Relation for Positive Token Propagation

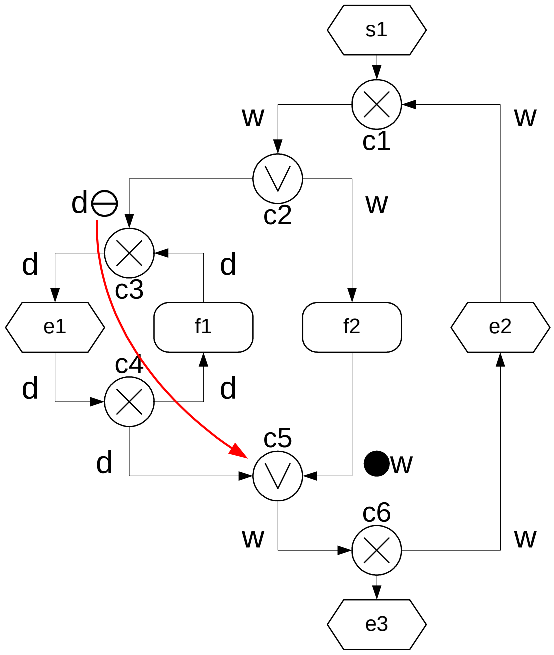


Figure 3.18: A structured EPC with a negative token on the negative upper corona of OR-join *c5*

since a node can fire only if all its outputs are empty. Due to the safeness property, we already know that the state space is finite since also the number of arcs is finite for an EPC. Another observation is that there are several state and context propagations that are not interesting to the user of the model. Therefore, the following section will make a distinction between the *transition relation* of an EPC that covers all state and context changes, and the *reachability graph* that only covers the propagation of positive tokens and hides context and negative token propagation.

This semantics definition based on state and context implies that the examples of Section 3.4 behave as follows.

- Figure 3.6(a): The single OR-join on the loop produces a wait context at $a9$. Therefore, it is blocked.
- Figure 3.6(b): The two OR-joins produce a wait context at $a23$ and $a24$. Therefore, they are both blocked.

- Figure 3.7: The three OR-joins are blocked due to a wait context at $a7$, $a14$, and $a21$.
- Figure 3.8(a): The OR-join $c2$ must wait for the second token on *a7*.
- Figure 3.8(b): The OR-join $c2$ must wait for the second token on *a7f*.
- Figure 3.9(a): The OR-join *c1* must wait for the token on *a7*.
- Figure 3.9(b): The OR-join *c1* must wait for the token on *a7*. The OR-split $c3a$ produces a negative token on $a7c$ so that $c3b$ can fire.

It can be seen that the refined EPCs exhibit the expected behavior similar to the unrefined cases, i.e. the OR-join in the structured block does not deadlock. Furthermore, if there is an OR-join as an entry point to a loop, it will deadlock if there is not a second XOR-entry that can propagate a token into this loop.

3.4.5 Transition Relation and Reachability Graph of EPCs

In this section, we formalize the concepts that were introduced in the previous section. In particular, we define the transition relations for each phase and the reachability graph of EPCs based on markings, i.e. state and context mappings σ and κ collectively. The reachability graph hides the transitions of the context propagation and negative token propagation phases. First, we provide definitions for marking, initial marking, and final marking. Then, we define the transition relations R^d, R^w, R^{-1}, and R^{+1} of an EPC for each of the four phases. Finally, we define the reachability graph RG based on the transition relations and an algorithm to calculate RG. *Please note that all definitions are applicable for relaxed syntactically correct EPCs* (see Definition 3.8 on page 47).

Definition of Initial and Final Marking

In this paragraph we define the sets of the initial and the final markings of an EPC similar to the definition in *Rump* [Rum99]. An initial marking is an assignment of positive or negative tokens to all start arcs while all other arcs have no token, and in a final marking only end arcs may hold positive tokens.

Definition 3.14 (Initial Marking of an EPC)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC and M_{EPC} its marking space. $I_{EPC} \subseteq M_{EPC}$ is defined as the set of all possible initial markings, i.e. $m \in I_{EPC}$ if and only if [6]:

- $\exists a_s \in A_s : \sigma_m(a_s) = +1$,
- $\forall a_s \in A_s$: $\sigma_m(a_s) \in \{-1, +1\}$,
- $\forall a_s \in A_s$: $\kappa_m(a_s) = wait$ if $\sigma_m(a_s) = +1$ and $\kappa_m(a_s) = dead$ if $\sigma_m(a_s) = -1$, and
- $\forall a \in A_{int} \cup A_e : \kappa_m(a) = wait$ and $\sigma_m(a) = 0$.

Definition 3.15 (Final Marking of an EPC)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC and M_{EPC} its marking space. $O_{EPC} \subseteq M_{EPC}$ is defined as the set of all possible final markings, i.e. $m \in O_{EPC}$ if and only if:

- $\exists a_e \in A_e$: $\sigma_m(a_e) = +1$ and
- $\forall a \in A_s \cup A_{int} : \sigma_m(a) \leq 0$.

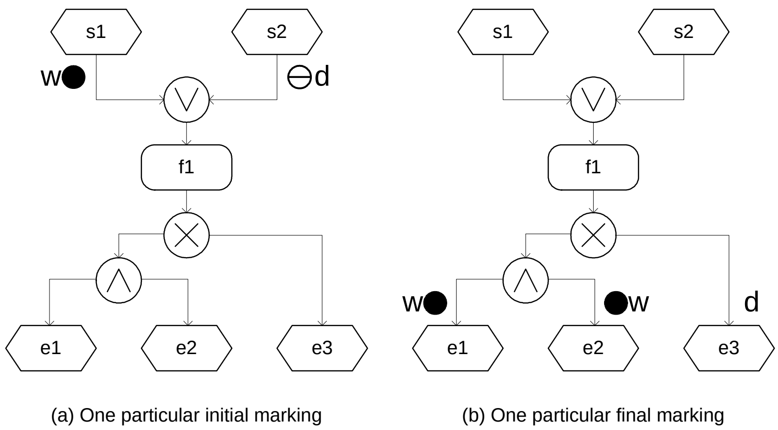


Figure 3.19: Initial and final marking of an EPC

Initial and final markings are the start and end points for calculating the transition relation of an EPC. Figure 3.19(a) illustrates one particular initial marking $i \in I$ which

[6] Note that the marking is given in terms of arcs. Intuitively, one can think of start events holding positive or negative tokens. However, the corresponding arc will formally represent this token.

assigns a positive token to the left start arc and a negative token to the right start arc. The OR-join synchronizes both these tokens and may produce (after some steps) the marking that is depicted in Figure 3.19(b). There, the left branch of the XOR-split has been taken which results in positive tokens on the end arcs after the AND-split and a dead context on the right end arc.

Phase 1: Transition Relation for Dead Context Propagation

Given these definitions related to the marking of an EPC, we define the transition relations for each phase. We can summarize the different rules of Figure 3.12 in a single one: if one input arc of the respective node has a dead context, then this is propagated to the output arcs.

Definition 3.16 (Transition Relations for Dead Context Propagation)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC, $N = E \cup F \cup C$ its set of nodes, and M_{EPC} its marking space. Then $R^d \subseteq M_{EPC} \times N \times M_{EPC}$ is the transition relation for dead context propagation and $(m, n, m') \in R^d$ if and only if:

$$
\begin{aligned}
&(\exists_{a \in n_{in}} : \kappa_m(a) = dead) \wedge \\
&(\forall_{a \in A} : \sigma_m(a) = \sigma_{m'}(a)) \wedge \\
&(\exists_{X \neq \emptyset} : X = \{a \in n_{out} \mid \sigma_m(a) = 0 \ \wedge \kappa_m(a) = wait\} \wedge \\
&\qquad (\forall_{a \in X} : \kappa_{m'}(a) = dead) \wedge \\
&\qquad (\forall_{a \in A \setminus X} : \kappa_{m'}(a) = \kappa_m(a))
\end{aligned}
$$

Furthermore, we define the following notations:

- $m_1 \overset{n}{\underset{d}{\rightarrow}} m_2$ if and only if $(m_1, n, m_2) \in R^d$. We say that in the dead context propagation phase marking m_1 enables node n and its firing results in m_2.
- $m \underset{d}{\rightarrow} m'$ if and only if $\exists n : m_1 \overset{n}{\underset{d}{\rightarrow}} m_2$.
- $m \overset{\tau}{\underset{d}{\rightarrow}} m'$ if and only if $\exists_{n_1,\ldots,n_q,m_1,\ldots,m_{q+1}} : \tau = n_1 n_2 \ldots n_q \in N* \ \wedge$ $m_1 = m \wedge m_{q+1} = m' \wedge m_1 \overset{n_1}{\underset{d}{\rightarrow}} m_2, m_2 \overset{n_2}{\underset{d}{\rightarrow}} \ldots \overset{n_q}{\underset{d}{\rightarrow}} m_{q+1}$.
- $m \overset{*}{\underset{d}{\rightarrow}} m'$ if and only if $\exists_\tau : m \overset{\tau}{\underset{d}{\rightarrow}} m'$.

- $m \overset{max}{\underset{d}{\rightarrow}} m'$ if and only if $\exists_\tau : m \overset{\tau}{\underset{d}{\rightarrow}} m' \land \nexists_{m'' \neq m'} : m' \underset{d}{\rightarrow} m''$.
- $max_d : M_{EPC} \rightarrow M_{EPC}$ such that $max_d(m) = m'$ if and only if $m \overset{max}{\underset{d}{\rightarrow}} m'$. The existence of a unique $max_d(m)$ is the subject of Theorem 3.1 below.

Theorem 3.1 (Dead Context Propagation terminates)**.** *For an EPC and a given marking m, there exists a unique $max_d(m)$ which is determined in a finite number of propagation steps.*

Proof. Regarding *uniqueness*, by contradiction: Consider an original marking $m_0 \in M_{EPC}$ and two markings $m_{max1}, m_{max2} \in M_{EPC}$ such that $m_0 \overset{max}{\underset{d}{\rightarrow}} m_{max1}$, $m_0 \overset{max}{\underset{d}{\rightarrow}} m_{max2}$, and $m_{max1} \neq m_{max2}$. Since both m_{max1} and m_{max2} can be produced from m_0 they share at least those arcs with dead context that were already dead in m_0. Furthermore, following from the inequality, there must be an arc a that has a wait context in one marking, but not in the other. Let us assume that this marking is m_{max1}. But if $\exists \tau : m_0 \overset{\tau}{\underset{d}{\rightarrow}} m_{max2}$ such that $\kappa_{m_{max2}}(a) = dead$, then there must also $\exists \tau' : m_{max1} \overset{\tau'}{\underset{d}{\rightarrow}} m'$ such that $\kappa_{m'}(a) = dead$ because m_{max2} is produced applying the propagation rules without ever changing a dead context to a wait context. Accordingly, there are further propagation rules that can be applied on m_{max1} and the assumption $m_0 \overset{max}{\underset{d}{\rightarrow}} m_{max1}$ is wrong. Therefore, if there are two m_{max1} and m_{max2}, they must have the same set of arcs with dead context, and therefore also the same set of arcs with wait context. Since both their states are equal to the state of m_0 they are equivalent, i.e., $max_d(m)$ is unique.
Regarding *finiteness*: Following Definition 3.11 on page 50, the number of nodes of an EPC is finite, and therefore the set of arcs is also finite. Since the number of dead context arcs is increased in each propagation step, no new propagation rule can be applied, at the latest after each arc has a dead context. Accordingly, dead context propagation terminates at the latest after $|A|$ steps. □

Phase 2: Transition Relation for Wait Context Propagation

For the wait context propagation, we also distinguish two cases based on the different transition relations of Figure 3.13. The first case covers (a) function, (b) intermediate event, (c) split, (d) and-join nodes. If the node belongs to this group and all input arcs are in a wait context, then the wait context is propagated to those output arcs that have a dead context and no state token on them. The second case, if the node is an XOR-join or an OR-join and one of the input arcs is in a wait context, then this is propagated to the dead output arc.

Definition 3.17 (Transition Relations for Wait Context Propagation)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC, $N = E \cup F \cup C$ its set of nodes, and M_{EPC} its marking space. Then $R^w \subseteq M_{EPC} \times N \times M_{EPC}$ is the transition relation for wait context propagation and $(m, n, m') \in R^w$ if and only if:

$$
\begin{aligned}
&((n \in F \cup E_{int} \cup S \cup J_{and}) \wedge \\
&(\forall_{a \in n_{in}} : \kappa_m(a) = wait) \wedge \\
&(\forall_{a \in A} : \sigma_m(a) = \sigma_{m'}(a)) \wedge \\
&(\exists_{X \neq \emptyset} : X = \{a \in n_{out} \mid \sigma_m(a) = 0 \ \wedge \kappa_m(a) = dead\} \wedge \\
&\qquad (\forall_{a \in X} : \kappa_{m'}(a) = wait) \wedge \\
&\qquad (\forall_{a \in A \setminus X} : \kappa_{m'}(a) = \kappa_m(a))) \\
&\vee \\
&((n \in J_{xor} \cup J_{or}) \wedge \\
&(\exists_{a \in n_{in}} : \kappa_m(a) = wait) \wedge \\
&(\forall_{a \in A} : \sigma_m(a) = \sigma_{m'}(a)) \wedge \\
&(\exists_{X \neq \emptyset} : X = \{a \in n_{out} \mid \sigma_m(a) = 0 \ \wedge \kappa_m(a) = dead\} \wedge \\
&\qquad (\forall_{a \in X} : \kappa_{m'}(a) = wait) \wedge \\
&\qquad (\forall_{a \in A \setminus X} : \kappa_{m'}(a) = \kappa_m(a)))
\end{aligned}
$$

Furthermore, we define the following notations:

- $m_1 \overset{n}{\underset{w}{\rightarrow}} m_2$ if and only if $(m_1, n, m_2) \in R^w$. We say that in the wait context propagation phase marking m_1 enables node n and its firing results in m_2.
- $m \underset{w}{\rightarrow} m'$ if and only if $\exists n : m_1 \overset{n}{\underset{w}{\rightarrow}} m_2$.
- $m \overset{\tau}{\underset{w}{\rightarrow}} m'$ if and only if $\exists_{n_1,...,n_q,m_1,...,m_{q+1}} : \tau = n_1 n_2 ... n_q \in N* \ \wedge$ $m_1 = m \wedge m_{q+1} = m' \wedge m_1 \overset{n_1}{\underset{w}{\rightarrow}} m_2, m_2 \overset{n_2}{\underset{w}{\rightarrow}} ... \overset{n_q}{\underset{w}{\rightarrow}} m_{q+1}$.
- $m \overset{*}{\underset{w}{\rightarrow}} m'$ if and only if $\exists_\tau : m \overset{\tau}{\underset{w}{\rightarrow}} m'$.

- $m \overset{max}{\underset{w}{\rightarrow}} m'$ if and only if $\exists_\tau : m \overset{\tau}{\underset{w}{\rightarrow}} m' \land \nexists_{m'' \neq m'} : m' \underset{w}{\rightarrow} m''$.
- $max_w : M_{EPC} \rightarrow M_{EPC}$ such that $max_w(m) = m'$ if and only if $m \overset{max}{\underset{w}{\rightarrow}} m'$. The existence of a unique $max_w(m)$ is the subject of Theorem 3.2 below.

Theorem 3.2 (Wait Context Propagation terminates)**.** *For an EPC and a given marking m, there exists a unique $max_w(m)$ which is determined in a finite number of propagation steps.*

Proof. Analogous proof as for Theorem 3.1. □

Phase 3: Transition Relation for Negative State Propagation

The transition rules for the various node types in this phase can be easily summarized in one transition relation: if all input arcs carry a negative token and all output arcs hold no negative or positive token, then consume all negative tokens on the input arcs and produce negative tokens on each output arc.

Definition 3.18 (Transition Relations for Negative State Propagation)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC, $N = E \cup F \cup C$ its set of nodes, and M_{EPC} its marking space. Then $R^{-1} \subseteq M_{EPC} \times N \times M_{EPC}$ is the transition relation for negative state propagation and $(m, n, m') \in R^{-1}$ if and only if:

$$\begin{aligned}
&(\forall_{a \in n_{in}} : \sigma_m(a) = -1) \land \\
&(\forall_{a \in n_{out}} : \sigma_m(a) = 0) \land \\
&(\forall_{a \in n_{in}} : \sigma_{m'}(a) = 0) \land \\
&(\forall_{a \in n_{out}} : \sigma_{m'}(a) = -1) \land \\
&(\forall_{a \in A \setminus n_{out}} : \kappa_{m'}(a) = \kappa_m(a)) \land \\
&(\forall_{a \in n_{out}} : \kappa_{m'}(a) = dead) \land \\
&(\forall_{a \in A \setminus (n_{in} \cup n_{out})} : \sigma_{m'}(a) = \sigma_m(a))
\end{aligned}$$

Furthermore, we define the following notations:

- $m_1 \overset{n}{\underset{-1}{\rightarrow}} m_2$ if and only if $(m_1, n, m_2) \in R^{-1}$. We say that in the negative state propagation phase marking m_1 enables node n and its firing results in m_2.

- $m \underset{-1}{\rightarrow} m'$ if and only if $\exists n : m_1 \overset{n}{\underset{-1}{\rightarrow}} m_2$.
- $m \overset{\tau}{\underset{-1}{\rightarrow}} m'$ if and only if $\exists_{n_1,...,n_q,m_1,...,m_{q+1}} : \tau = n_1 n_2 ... n_q \in N* \wedge$ $m_1 = m \wedge m_{q+1} = m' \wedge m_1 \overset{n_1}{\underset{-1}{\rightarrow}} m_2, m_2 \overset{n_2}{\underset{-1}{\rightarrow}} ... \overset{n_q}{\underset{-1}{\rightarrow}} m_{q+1}$.
- $m \overset{*}{\underset{-1}{\rightarrow}} m'$ if and only if $\exists_\tau : m \overset{\tau}{\underset{-1}{\rightarrow}} m'$.
- $m \overset{max}{\underset{-1}{\rightarrow}} m'$ if and only if $\exists_\tau : m \overset{\tau}{\underset{-1}{\rightarrow}} m' \wedge \nexists_{m'' \neq m'} : m' \underset{-1}{\rightarrow} m''$.
- $max_{-1} : M_{EPC} \rightarrow M_{EPC}$ such that $max_{-1}(m) = m'$ if and only if $m \overset{max}{\underset{-1}{\rightarrow}} m'$. The existence of a unique $max_{-1}(m)$ is discussed below in Theorem 3.3.

Theorem 3.3 (Negative State Propagation terminates)**.** *For an EPC and a given marking m, there exists a unique $max_{-1}(m)$ which is determined in a finite number of propagation steps.*

Proof. Regarding finiteness, by contradiction. Since an EPC is safe, i.e. there is at maximum one token per arc, it is a prerequisite for an infinite propagation that there is a cyclic structure in the process in which the negative token runs into an infinite loop. Due to the coherence property of an EPC, and the minimum number of one start and one end node (Definition 3.11), two cases of a cyclic path can be distinguished:

(i) cyclic path $a \hookrightarrow a$ with $\nexists e \in E_s : e \hookrightarrow a$: in this case the loop could potentially propagate a negative token infinitely, but it will never receive a token since there is no path from a start node into the cyclic path. Furthermore, relaxed syntactically correct EPCs do not contain such paths according to Definition 3.8.

(ii) cyclic path $a \hookrightarrow a$ with $\exists e \in E_s : e \hookrightarrow a$: In this case, there must be a join j on a cyclic path $a \hookrightarrow a$ such that there exists an arc (x, j) and there is no path $a \hookrightarrow x$. Therefore, a negative token could only be propagated infinitely on the path $a \hookrightarrow a$ if the join j would receive repeatedly ad infinitum negative tokens on the arc (x, j) in order to allow j to fire according to Definition 3.18. Since the number of tokens on arcs is limited to one, this is only possible if there is another cyclic path $b \hookrightarrow b$ that produces negative tokens ad infinitum on a split node s. Again, for this cyclic path $b \hookrightarrow b$, the two cases (i) and (ii) can be distinguished. Accordingly, there must be another cyclic path $c \hookrightarrow c$ that feeds the path with b, and so forth.

Since the existence of a cyclic path that propagates negative tokens infinitely depends on the existence of another such path, there is a contradiction. □

Regarding uniqueness we do not provide a formal proof here. Consider that there exist an original marking $m_0 \in M_{EPC}$ and two markings $m_{max1}, m_{max2} \in M_{EPC}$ such that $m_0 \overset{max}{\underset{-1}{\rightarrow}} m_{max1}$, $m_0 \overset{max}{\underset{-1}{\rightarrow}} m_{max2}$, and $m_{max1} \neq m_{max2}$. According to the transition relation, there are no transitions that could compete for tokens such as in non free-choice Petri nets, i.e. the firing of a transition cannot disable another one, and there are no alternative transitions for an enabled node. Furthermore, a context change of an arc has no impact on the applicability of a rule and no positive tokens are involved in firings. Therefore, m_{max1} and m_{max2} must either be equivalent or there must be a transition enabled in one of them such that the max property of it does not hold.

Phase 4: Transition Relation for Positive State Propagation

For OR-joins, we already described the concept of a negative upper corona in Section 3.4.4 on page 78. The firing of an OR-join consumes not only the negative tokens on its input arcs, but also the negative tokens on its negative upper corona. This way, no unnecessary negative tokens remain in the EPC.

Definition 3.19 (Dead Empty Path, Negative Upper Corona)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC, $N = E \cup F \cup C$ its set of nodes, and a marking $m \in M_{EPC}$. Then, we define the negative upper corona of a node $n \in N$ based on a dead empty path. A *dead empty path* $a \overset{d}{\underset{m}{\hookrightarrow}} b$ refers to a sequence of nodes $n_1, \ldots, n_k \in N$ with $a = n_1$ and $b = n_k$ such that for $(n_1, n_2) \in A : \sigma_m(n_1, n_2) = -1$ and $\forall i \in 2, \ldots, k-1$ holds: $(n_i, n_{i+1}) \in A \wedge \sigma_m(n_i, n_{i+1}) = 0 \wedge \kappa_m(n_i, n_{i+1}) = dead$. Then, the *negative upper corona* $\overset{-1}{\underset{m}{\hookrightarrow}} n = \{a \in A | a = (s, t) \wedge \sigma(a) = -1 \wedge t \overset{d}{\underset{m}{\hookrightarrow}} n\}$ refers to those arcs with a negative token whose target node t is a transitive predecessor of n and has a dead empty path to n in marking m.

The transition rules for the various node types can be easily summarized as follows: (1) for function, event, and AND-connector nodes, positive tokens on all input arcs are consumed and propagated to all output arcs, if all of them are empty. The input context is set to dead and the output context to wait. (2) For XOR-connectors, one input token is consumed from one input arc and propagated to one of the output arcs if all of them are empty. The respective input arc is set to a dead context, as well as those output arcs that do not receive the token. The output arc with the positive token gets a wait context. (3) For OR-splits, the positive token is consumed from the input, and a combination of positive and negative tokens is produced at the output arcs such that at least one positive token is available. Furthermore, each output arc with a positive token gets a wait context while the others get a dead context. (4) OR-joins fire either if all input arcs are not empty and one of them has a positive token, or if there is no empty arc with a wait context and at least one positive token on the inputs. Then, all input tokens are consumed, plus potentially negative tokens on the negative upper corona, the input arcs are set to a dead context, and a positive token is produced on the output with a wait context.

Definition 3.20 (Transition Relation for Positive State Propagation)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC, $N = E \cup F \cup C$ its set of nodes, and M_{EPC} its marking space. Then $R^{+1} \subseteq M_{EPC} \times N \times M_{EPC}$ is the transition relation for positive state propagation and $(m, n, m') \in R^{+1}$ if and only if:

$$
\begin{aligned}
&((n \in F \cup E_{int} \cup C_{and}) \wedge \\
&(\forall_{a \in n_{in}} : \sigma_m(a) = +1) \wedge \\
&(\forall_{a \in n_{out}} : \sigma_m(a) = 0) \wedge \\
&(\forall_{a \in n_{in}} : \sigma_{m'}(a) = 0 \ \wedge \ \kappa_{m'}(a) = dead) \wedge \\
&(\forall_{a \in n_{out}} : \sigma_{m'}(a) = +1 \ \wedge \ \kappa_{m'}(a) = wait) \wedge \\
&(\forall_{a \in A \setminus (n_{in} \cup n_{out})} : \kappa_{m'}(a) = \kappa_m(a)) \wedge \\
&(\forall_{a \in A \setminus (n_{in} \cup n_{out})} : \sigma_{m'}(a) = \sigma_m(a))) \\
&\vee \\
&((n \in C_{xor}) \wedge \\
&(\exists_{a_1 \in n_{in}} : (\sigma_m(a_1) = +1 \wedge \sigma_{m'}(a_1) = 0 \wedge \\
&\qquad \kappa_m(a_1) = wait \ \wedge \ \kappa_{m'}(a_1) = dead) \wedge \\
&\qquad (\forall_{a \in n_{out}} : \sigma_m(a) = 0) \wedge \\
&\qquad (\exists_{X \wedge a_2 \in n_{out}} : X = \{a \in n_{in} \mid \sigma_m(a) = -1 \ \wedge \kappa_m(a) = dead\} \wedge \\
&\qquad\qquad (\sigma_{m'}(a_2) = +1 \ \wedge \ \kappa_{m'}(a_2) = wait) \wedge \\
&\qquad\qquad (\forall_{a \in A \setminus \{a_1, a_2\}} : \kappa_{m'}(a) = \kappa_m(a)) \wedge
\end{aligned}
$$

$$
\begin{aligned}
&\qquad\qquad (\forall_{a \in X} : \sigma_{m'}(a) = 0 \ \wedge \ \kappa_{m'}(a) = \kappa_m(a)) \wedge \\
&\qquad\qquad (\forall_{a \in A \setminus (X \cup \{a_1, a_2\})} : \sigma_{m'}(a) = \sigma_m(a)))))\\
&\vee \\
&((n \in S_{or}) \wedge \\
&(\forall_{a \in n_{in}} : \sigma_m(a) = +1) \wedge \\
&(\forall_{a \in n_{out}} : \sigma_m(a) = 0) \wedge \\
&(\forall_{a \in n_{in}} : \sigma_{m'}(a) = 0 \ \wedge \kappa_{m'}(a) = dead) \wedge \\
&(\exists_{X \neq \emptyset} : X = \{a \in n_{out} \mid \sigma_{m'}(a) = +1 \ \wedge \kappa_{m'}(a) = wait\} \wedge \\
&\qquad (\forall_{a \in n_{out} \setminus X} : \sigma_{m'}(a) = -1 \ \wedge \kappa_{m'}(a) = dead) \wedge \\
&\qquad (\forall_{a \in A \setminus (n_{in} \cup n_{out})} : \kappa_{m'}(a) = \kappa_m(a) \wedge \sigma_{m'}(a) = \sigma_m(a)))\\
&\vee \\
&((n \in J_{or}) \wedge \\
&(\exists_{X \neq \emptyset} : X = \{a \in n_{in} \mid \sigma_m(a) = +1 \ \wedge \kappa_m(a) = wait\}) \wedge \\
&(\exists_{Y} : Y = \{a \in n_{in} \mid \sigma_m(a) = -1 \ \wedge \kappa_m(a) = dead\}) \wedge \\
&(\exists_{Z} : Z = \{a \in n_{in} \mid \sigma_m(a) = 0 \ \wedge \kappa_m(a) = dead\}) \wedge \\
&(X \cup Y \cup Z = n_{in}) \wedge \\
&(\forall_{a \in n_{out}} : \sigma_m(a) = 0) \wedge \\
&(\forall_{a \in n_{in}} : \sigma_{m'}(a) = 0 \wedge \kappa_{m'}(a) = dead)) \wedge \\
&(\forall_{a \in n_{out}} : \sigma_{m'}(a) = +1 \wedge \kappa_{m'}(a) = wait) \wedge \\
&(\exists_{U \subset A} : U = \overset{-1}{\underset{m}{\hookrightarrow}} n \wedge \\
&\qquad (\forall_{a \in U} : \sigma_{m'}(a) = 0 \wedge \kappa_{m'}(a) = \kappa_m(a)) \wedge \\
&\qquad (\forall_{a \in A \setminus (U \cup n_{in} \cup n_{out})} : \sigma_{m'}(a) = \sigma_m(a) \wedge \kappa_{m'}(a) = \kappa_m(a)))).
\end{aligned}
$$

Furthermore, we define the following notations:

- $m_1 \overset{n}{\underset{+1}{\rightarrow}} m_2$ if and only if $(m_1, n, m_2) \in R^{+1}$. We say that in the positive state propagation phase marking m_1 enables node n and its firing results in m_2.
- $m \underset{+1}{\rightarrow} m'$ if and only if $\exists n : m_1 \overset{n}{\underset{+1}{\rightarrow}} m_2$.
- $m \overset{\tau}{\underset{+1}{\rightarrow}} m'$ if and only if $\exists_{n_1, \ldots, n_q, m_1, \ldots, m_{q+1}} : \tau = n_1 n_2 \ldots n_q \in N* \ \wedge$ $m_1 = m \wedge m_{q+1} = m' \wedge m_1 \overset{n_1}{\underset{+1}{\rightarrow}} m_2, m_2 \overset{n_2}{\underset{+1}{\rightarrow}} \ldots \overset{n_q}{\underset{+1}{\rightarrow}} m_{q+1}$.

- $m \underset{+1}{\overset{*}{\rightarrow}} m'$ if and only if $\exists_\tau : m \underset{+1}{\overset{\tau}{\rightarrow}} m'$.

Since the transition relation covers several marking changes that are not interesting for an observer of the process, we define the reachability graph RG of an EPC in the following section. It includes only transitions of the positive state propagation phase.

Calculating the Reachability Graph for EPCs

In this section, we define the reachability graph of an EPC and present an algorithm to calculate it. First we formalize the concept of reachability related to an EPC.

Definition 3.21 (Reachability related to an EPC)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC, $N = E \cup F \cup C$ its set of nodes, and M_{EPC} its marking space. Then, a marking $m' \in M_{EPC}$ is called reachable from another marking m if and only if $\exists n \in N \;\wedge\; m_1, m_2, m_3 \in M_{EPC} : max_d(m) = m_1 \wedge max_w(m_1) = m_2 \wedge max_{-1}(m_2) = m_3 \wedge m_3 \underset{+1}{\overset{n}{\rightarrow}} m'$. Furthermore, we define the following notations:

- $m \overset{n}{\rightarrow} m'$ if and only if m' is reachable from m.
- $m \rightarrow m' \Leftrightarrow \exists n \in N : m \overset{n}{\rightarrow} m'$.
- $m \overset{\tau}{\rightarrow} m'$ if and only if $\exists_{n_1,\ldots,n_q,m_1,\ldots,m_{q+1}} : \tau = n_1 n_2 \ldots n_q \in N* \;\wedge$
  $m_1 = m \wedge m_{q+1} = m' \wedge m_1 \overset{n_1}{\rightarrow} m_2, m_2 \overset{n_2}{\rightarrow} \ldots \overset{n_q}{\rightarrow} m_{q+1}$.
- $m_1 \overset{*}{\rightarrow} m_q \Leftrightarrow \exists \tau : m_1 \overset{\tau}{\rightarrow} m_q$.

Definition 3.22 (Reachability Graph of an EPC)**.** Let $EPC = (E, F, C, l, A)$ be a relaxed syntactically correct EPC, $N = E \cup F \cup C$ its set of nodes, and M_{EPC} its marking space. Then, the reachability graph $RG \subseteq M_{EPC} \times N \times M_{EPC}$ of an EPC contains the following nodes and transitions:

(i) $\forall m \in I_{EPC} : m \in RG$.

(ii) $(m, n, m') \in RG$ if and only if $m \overset{n}{\rightarrow} m'$.

The calculation of RG requires an EPC as input and a set of initial markings $I \subseteq I_{EPC}$. For several EPCs from practice, such a set of initial markings will not be available.

In this case, one can easily calculate the set of all possible initial markings. Algorithm 1 uses an object-oriented pseudo code notation to define the calculation. In particular, we assume that RG is an instance of the class $ReachabilityGraph$, $propagated$ an instances of class Set, and $toBePropagated$ an instance of class $Stack$ that provides the methods $pop()$ and $push()$. Furthermore, $currentMarking$, $oldMarking$, and $newMarking$ are instances of class $Marking$ that provides the methods $clone()$ to return a new, but equivalent marking, $propagateDeadContext(EPC)$, $propagateWaitContext(EPC)$, and $propagateNegativeTokens(EPC)$ to change the marking according to the transitions of the respective phase, i.e. to determine max_d, max_w, and max_{-1} of the current marking. Finally, $propagatePositiveTokens(EPC)$ returns a set of (node,marking) pairs including the node that can fire and the marking that is reached after the firing.

In lines 1-3, the sets RG and $propagated$ are initialized with the empty set, and the stack $toBePropagated$ is filled with all initial markings of the set I_{EPC}. The while loop between lines 4-18 calculates new markings for the marking that is on top of the stack $toBePropagated$. In particular, $currentMarking$ receives the top marking from the stack (line 5), and it is cloned into the $oldMarking$ object (line 6). In lines 7-9, the propagations of dead and wait context and of negative tokens are applied on $currentMarking$. Then, in line 10, the pairs of nodes and new markings that can be reached from the old marking are stored in the set $nodeNewMarking$. After that, the old marking is added to the $propagated$ set (line 11). In lines 12-17, for each pair of node and new marking a new transition $(oldMarking, node, newMarking)$ is added to RG. If a new marking has not yet been propagated, it is pushed on top of the $toBePropagated$ stack (lines 14-16). Using a stack, the reachability graph is calculated in a depth-first manner. Finally, in line 19 RG is returned.

3.4.6 Tool Support for the Novel EPC Semantics

Based on the previous algorithm, we have implemented the novel EPC semantics as a conversion plug-in for the *ProM* (Process Mining) framework [DMV+05, VDMA06, BHK+06]. ProM was originally developed as a tool for *process mining*, which is a domain that aims at extracting information from event logs to capture the business process as

Algorithm 1 Pseudo code for calculating the reachability graph of an EPC

Require: $EPC = (E, F, C, l, A), I \subseteq M$

```
RG ← ∅
toBePropagated ← I_EPC
propagated ← ∅
while toBePropagated ≠ ∅ do
    currentMarking ← toBePropagated.pop()
    oldMarking ← currentMarking.clone()
    currentMarking.propagateDeadContext(EPC)
    currentMarking.propagateWaitContext(EPC)
    currentMarking.propagateNegativeTokens(EPC)
    nodeNewMarking ← currentMarking.propagatePositiveTokens(EPC)
    propagated.add(oldMarking)
    for all (node, newMarking) ∈ nodeNewMarkings do
        RG.add(oldMarking, node, newMarking)
        if newMarking ∉ propagated then
            toBePropagated.push(newMarking)
        end if
    end for
end while
return RG
```

it is being executed (cf. e.g. [ADH+03, AWM04, CW98, GCC+04, Her00]). In the meantime, the functionality of ProM was extended to include other types of analysis, model conversions, model comparison, etc. This was enabled by the plug-able architecture of ProM, that allows to add new functionality without changing the framework itself, and the fact that ProM supports multiple modeling languages. Since ProM can interact with a variety of existing systems, e.g., *workflow management systems* such as Staffware, Oracle BPEL, Eastman Workflow, WebSphere, InConcert, FLOWer, Caramba, and YAWL, *simulation tools* such as ARIS, EPC Tools, Yasper, and CPN Tools, *ERP systems* like PeopleSoft and SAP, *analysis tools* such as AGNA, NetMiner, Viscovery, AlphaMiner, and ARIS PPM (cf. [BHK+06]), the plug-in for the new EPC semantics can easily be used for the analysis of existing models. Currently, there are more than 150 plug-ins in release 4.1. ProM basically supports five kinds of plug-ins:

Mining plug-ins to take a log and produce a model,

Import plug-ins to import a model from file, and possibly use a log to identify the relevant objects in the model,

Export plug-ins to export a model to file,

Conversion plug-ins to convert one model into another, and

Analysis plug-ins to analyze a model, potentially in combination with a log.

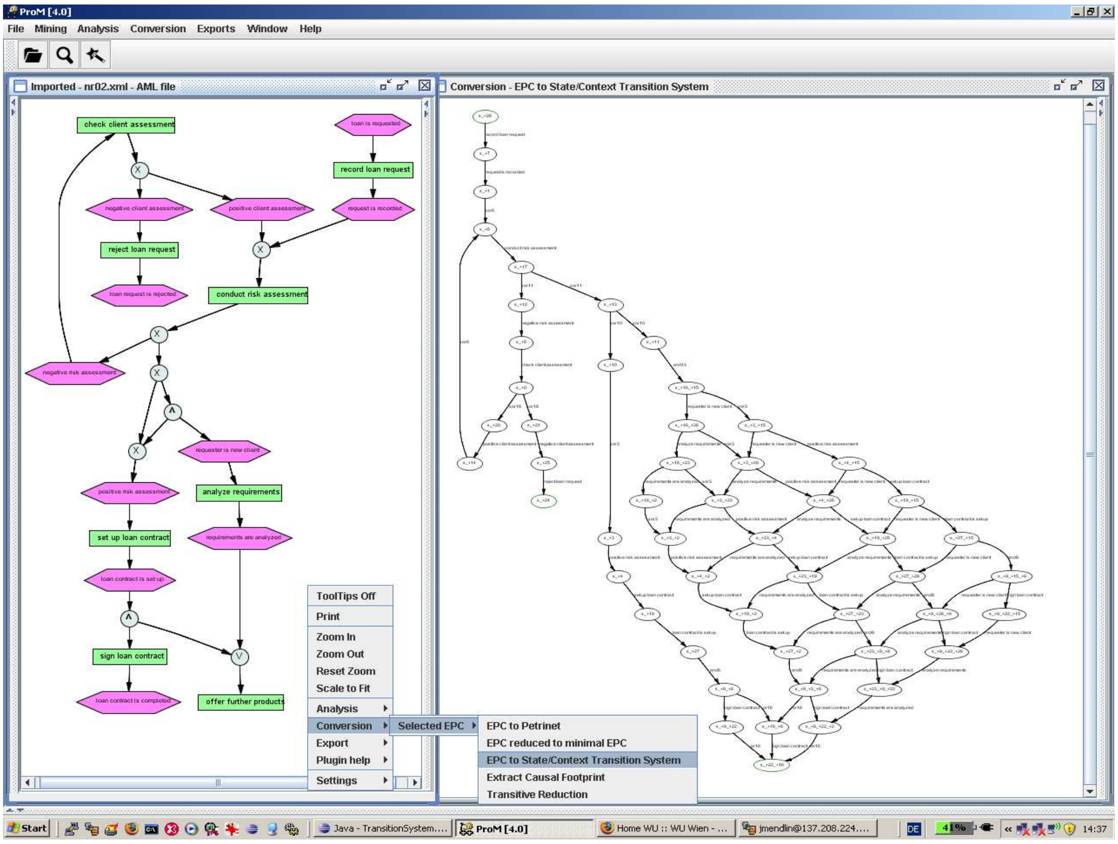


Figure 3.20: Calculating the reachability graph in ProM

The conversion plug-in maps an EPC to the transition systems package (cf. [ARD+06, RGA+06]) that was developed for an implementation of the incremental workflow mining approach by *Kindler, Rubin, and Schäfer* [KRS05, KRS06a, KRS06b]. Figure 3.20 illustrates how the conversion plug-in works. First, one has to load an EPC business process model into ProM, for instance, by using the import plug-in for the ARIS XML format [IDS03b] or for the EPC Markup Language [MN06]. In the figure, the EPC example model for a loan request process that we introduced in the beginning of this chapter is loaded. Since ProM generates a new layout automatically, the model looks different compared to the previous figure. Once the EPC is displayed in ProM, one can click on it, trigger the conversion plug-in "EPC to State/Context Transition System", and the reachability graph is calculated and shown in a new ProM window. The dense network of states and transitions on the right-hand side stems from the concurrent execution, if there is both a positive risk assessment for the loan request and the requester is a new customer. There are two markings that do not serve as a source for another transition in case if the request is rejected or accepted. Both these markings are displayed with a green border since they are proper final markings. If they were deadlocks, they would be drawn with a red border.

One of the nice features of the transition system package is that it provides an export to the file format of Petrify. *Petrify* is a software tool developed by *Cortadella, Kishinevsky, Lavagno, and Yakovlev* [CKLY98, Cor98] that can not only generate the state space for a Petri net, but also a Petri net from a transition system. The concepts of this Petri net synthesis builds on the theory of regions by *Ehrenfeucht and Rozenberg* [ER89, BD98]. Running Petrify with the reachability graph of the Loan Request example EPC of Figure 3.1 generates a free-choice Petri net as shown in Figure 3.21. It is interesting to see how the OR-join $or16$ is treated in the Petri net synthesis. It requires a token at each of the two input places before it can fire. If both the *positive risk assessment* and the *requester is new client* branch are executed, the OR-join synchronizes these paths via its two input places. If only the *positive risk assessment* branch is executed, the required tokens are produced by $xor3$. The decision point $xor11$ is the same as in the EPC model. Furthermore, it can be seen that each alternative of an XOR-split becomes a transition of its own (see $xor10$ and $xor10\_1$ or $xor11$ and $xor11\_1$) while the AND-split $and13$ remains one transition in the Petri net. The generation of a reachability graph for an EPC and the synthesis of a Petri net could be an important step to bring EPCs and Petri nets closer together. In particular, such a procedure could be a way to get rid of OR-joins for a Petri net implementation that has been modelled with EPCs in the design phase.

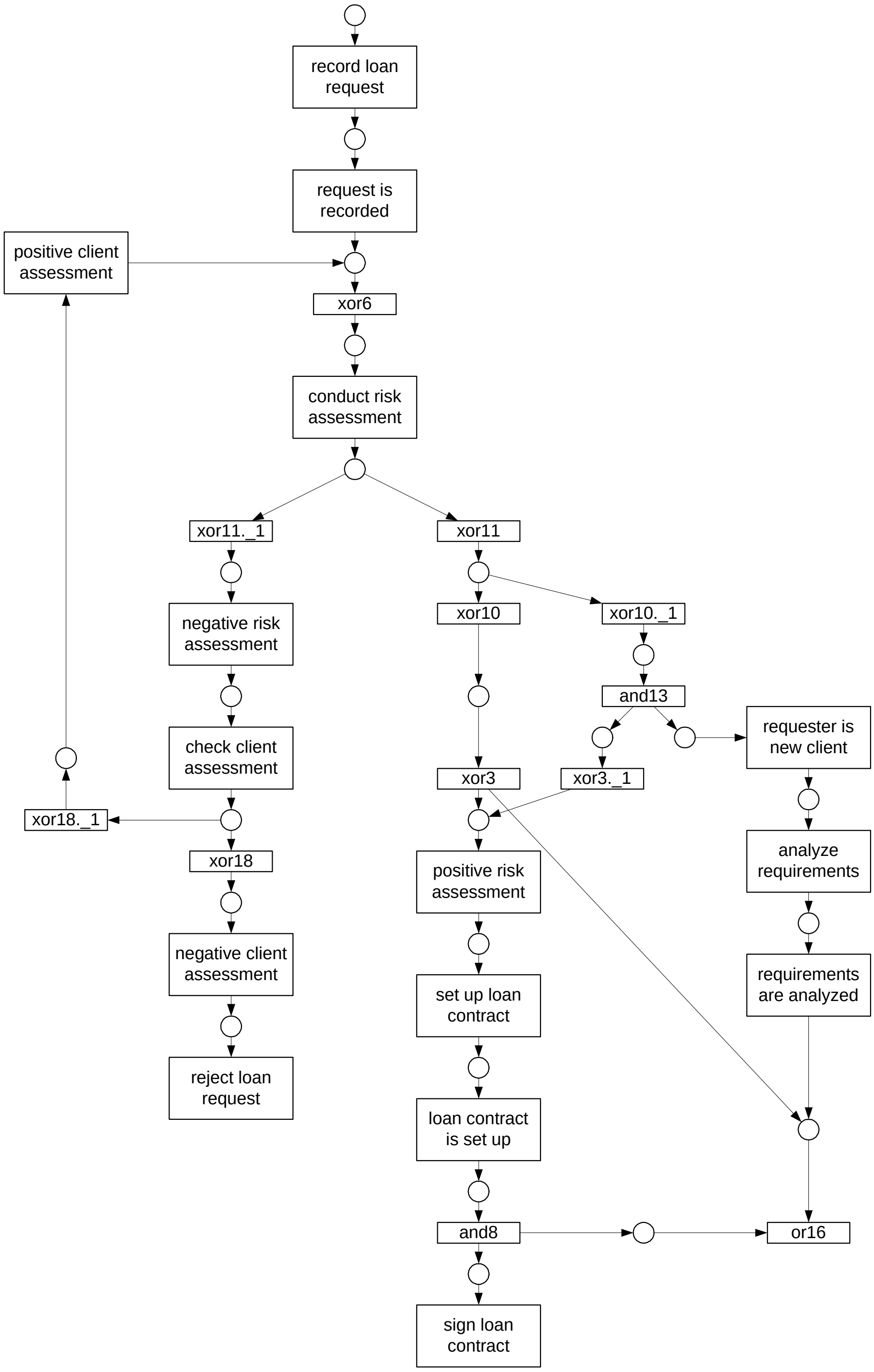


Figure 3.21: A Petri net that is bisimilar to the Loan Request EPC

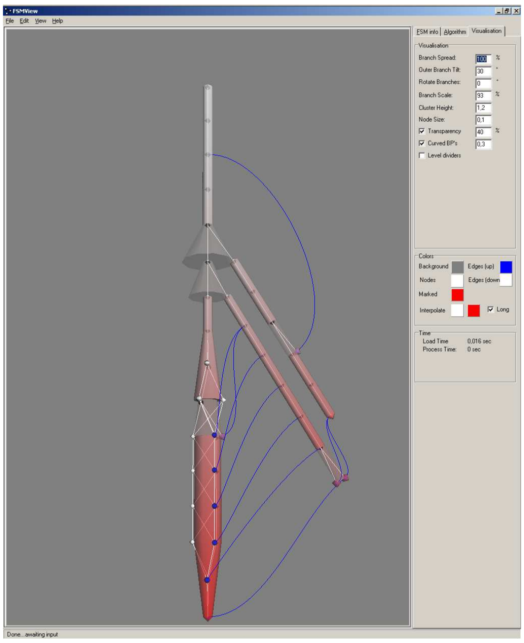


Figure 3.22: A visualization of the state space of the Loan Request Petri net

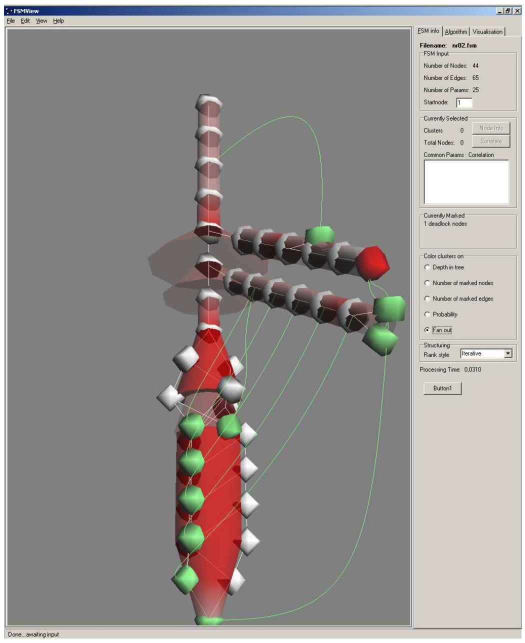


Figure 3.23: Another visualization of the Loan Request state space

Another useful application related to the ProM plug-in is the possibility to export to the FSM format via the Petri net analysis plug-in in ProM. This format can be loaded into the visualization tool FSMTool by *Groote and Van Ham* [HWW02, GH03, GH06]. FSMTool provides sophisticated interactive and customizable visualization of large state transition systems. The general visualization principle of FSMTool is to project the state space on levels of a backbone in such a way that structural symmetry can easily be seen. The Figures 3.22 and 3.23 visualize the state space of the Loan Request Petri net that was generated by Petrify as a three-dimensional backbone. The two decision points of this process are represented as cones in the upper part of the backbone. Each of these decision points splits off a new branch of execution that is visualized as a separate arm. On the first arm for negative risk assessment, there is a green line in Figure 3.22 (in Figure 3.23 it is blue) that represents an iteration of the loop. The other green lines highlight the activation of a node that is closer to the start node than the node that had control before. The thick pillar of the backbone represents the parallel execution after the AND-split. Overall, the FSMTool is a useful addition to the ProM plug-ins for understanding the complexity of the state space. Still, certain information about function labels is not present and there is no direct connection to the process model.

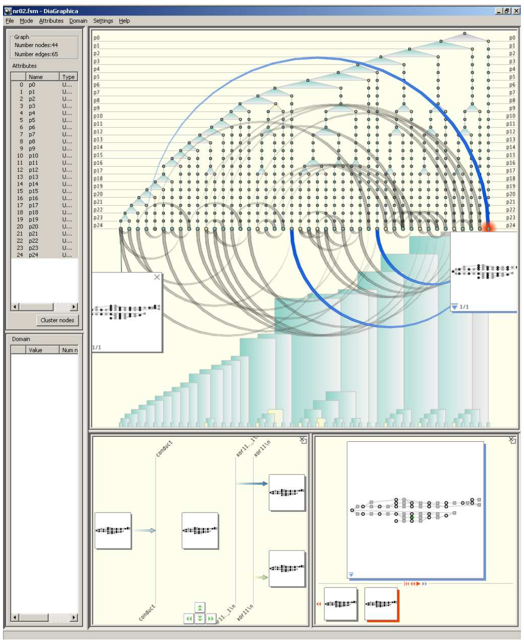


Figure 3.24: Visualization of the Petri net and the state space in DiaGraphica

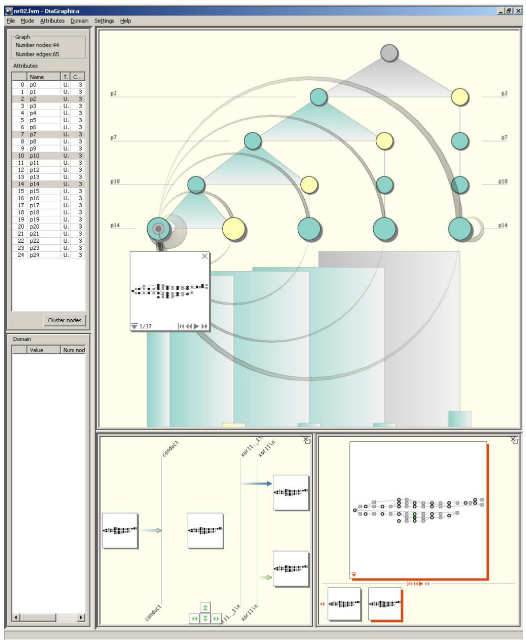


Figure 3.25: Clustering of places for the same state space in DiaGraphica

This shortcoming is the motivation of the work by *Verbeek, Pretorius, Van der Aalst, and Van Wijk* [VPAJ07] on a two-dimensional projection of state spaces as an extension to the Diagraphica tool of *Pretorius and Van Wijk* [PW05, PW06a]. Diagraphica can also load FSM files and in addition the diagram of a Petri net. Figure 3.24 shows that DiaGraphica uses an attribute clustering technique where, in this case, the attributes are related to the places of the Petri net. As Figure 3.25 shows, there may be multiple places in a cluster depending on the selections of the user. Transitions are represented as arcs. This figure permits an interesting observation. Below the diagonal line of yellow clusters the clustering hierarchy does not branch anymore. This means that for the selected places, only one can be marked at the same time (cf. [VPAJ07, p.16]). Further interpretations of different clustering patterns are discussed in [VPAJ07].

Based on the implementation of the reachability graph calculation in ProM, we can relate the novel EPC semantics to several other tools and approaches for analysis, synthesis, and visualization of process models and state spaces. This way, researchers can easily benefit from the EPC semantics and analyze its relationship to other formalisms.

3.5 EPCs and other Process Modeling Languages

In this section, we provide a comparison of EPCs with other business process modeling languages. The selection includes Workflow nets [Aal97], UML Activity Diagrams (UML AD) [OMG04], BPMN [OMG06], and YAWL [AH05], and is meant to illustrate differences and commonalities without going into mapping details. We first discuss whether these other process modeling languages offer elements similar to the different EPC connectors. After that, we utilize the workflow patterns documented in [AHKB03] to compare the languages. BPEL [CGK+02, ACD+03, AAB+05], which is also receiving increasing attention as a standard, is not included here since it addresses the execution rather than the conceptual modeling of processes. For further details on the relationship between EPCs and BPEL, refer to [MZ05a, ZM05, MZ05b, MLZ05, MLZ06b, MLZ06a]. For a workflow pattern analysis of BPEL, see [WADH03]. Furthermore, the XPDL standard [Wor02, Wor05] has also gained some support in the industry for the definition of executable workflow process. A workflow pattern analysis of XPDL is reported in [Aal03]. Other approaches for comparing process modeling languages are reported in [SAJ+02, RG02, BKKR03, Mue04, LK06].

3.5.1 Comparison based on Routing Elements

The six different connectors of EPCs, i.e., XOR-split and XOR-join, AND-split and AND-join, OR-split and OR-join, provide the means to model complex routing and ordering between activities of a business process. Table 3.4 takes these routing elements as a benchmark to compare EPCs with other business process modeling languages. It shows that the behavioral semantics of XOR-connectors and AND-connectors, as well as OR-split connectors, can be represented in all the considered languages. In *Workflow nets* XOR-connectors and AND-connectors are captured by places and transitions with multiple input and output arcs, respectively. OR-split behavior can be specified as

Appendix A

Errors found with *xoEPC*

This appendix shows those EPCs of the SAP Reference Model for which *xoEPC* found errors. The rest size is indicated in brackets. Please note that some models have up to nine problems being identified by *xoEPC*. Those models that are not completely reduced may still include errors that *xoEPC* did not find.

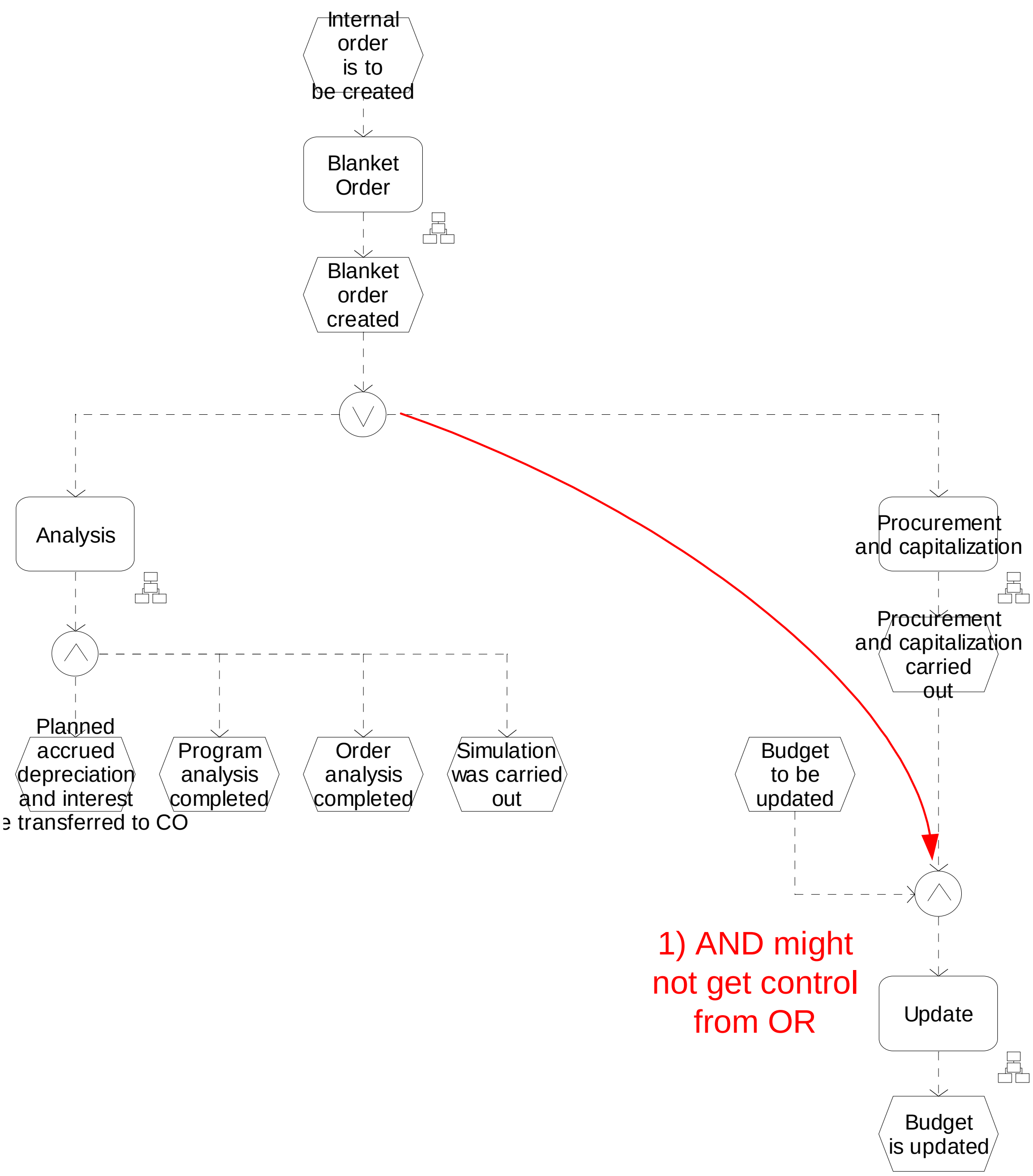


Figure A.1: Asset Accounting – Direct Capitalization (completely reduced)

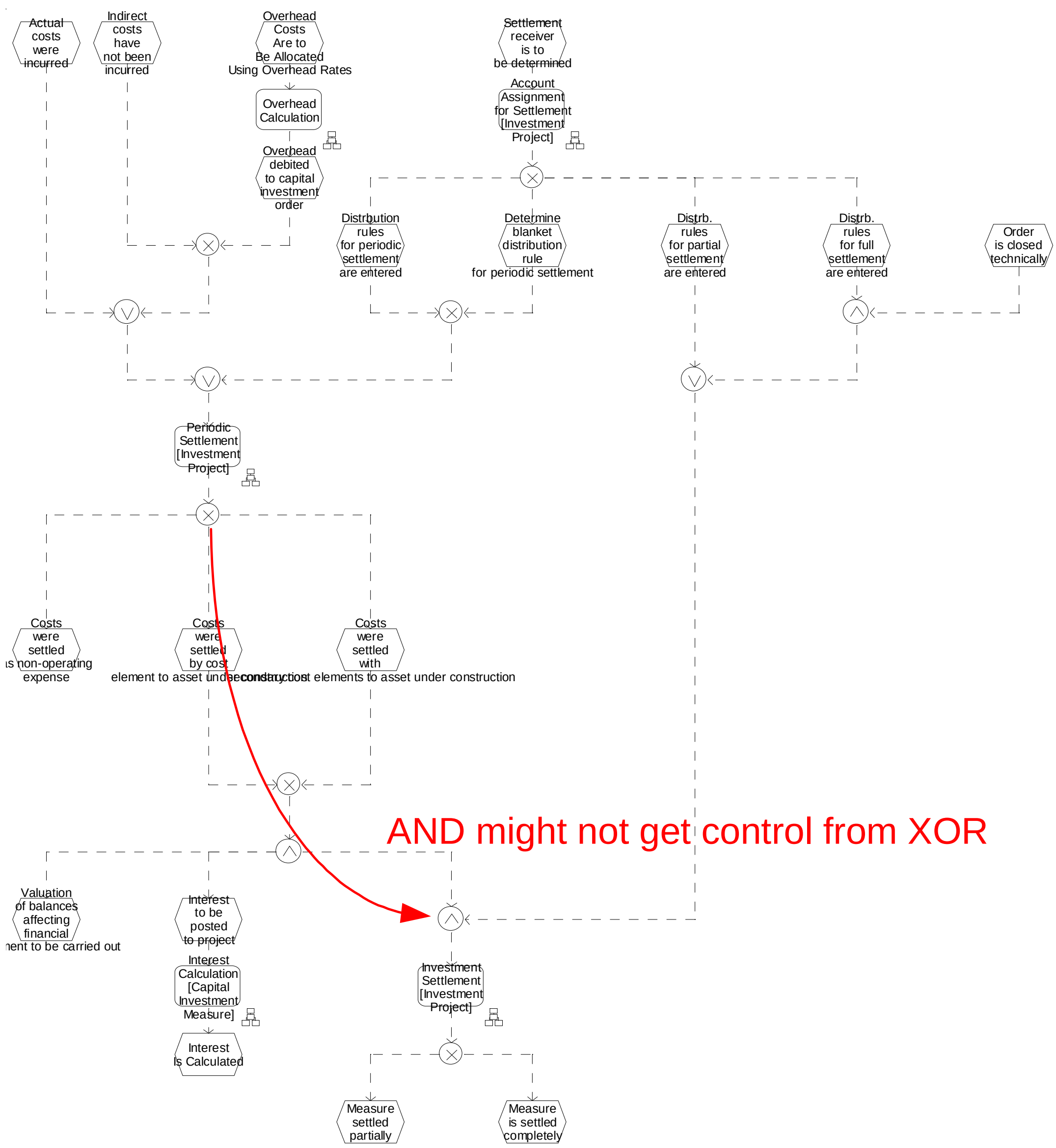


Figure A.2: Asset Accounting – Handling Complex Investment Measures – Period-End Closing and Settlement (reduced size 9)

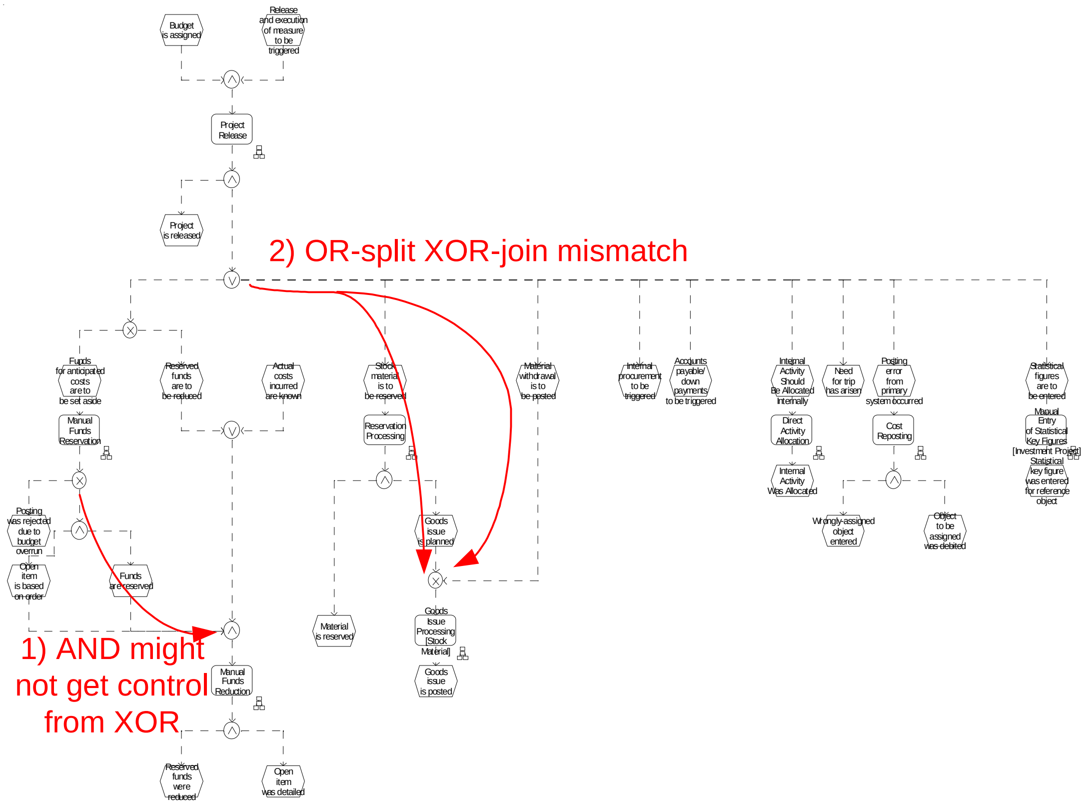


Figure A.3: Asset Accounting – Handling Complex Investment Measures – Release and Implementation of Measure (reduced size 8)

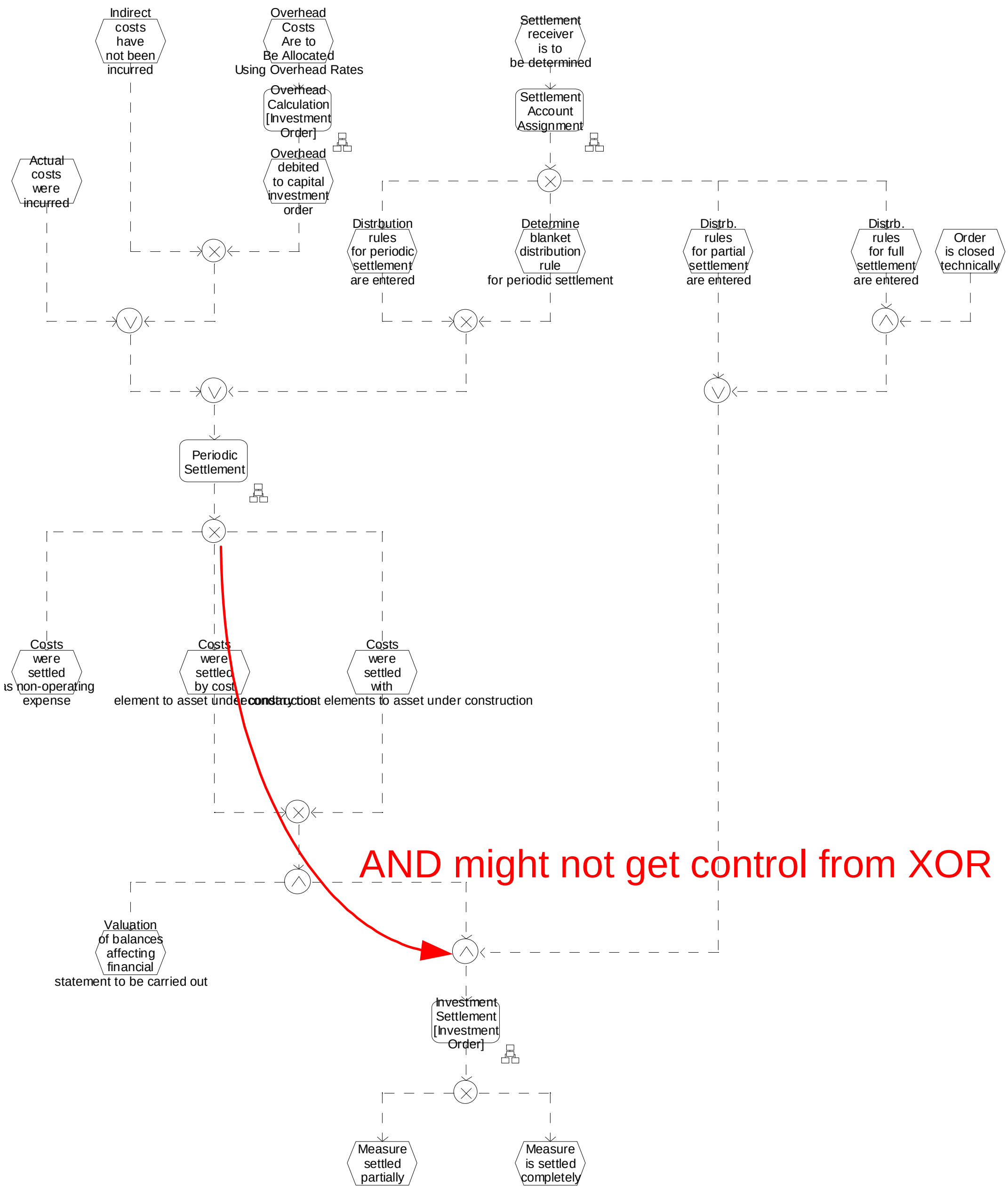


Figure A.4: Asset Accounting – Handling Simple Investment Measures – Period-End Closing and Settlement (reduced size 9)

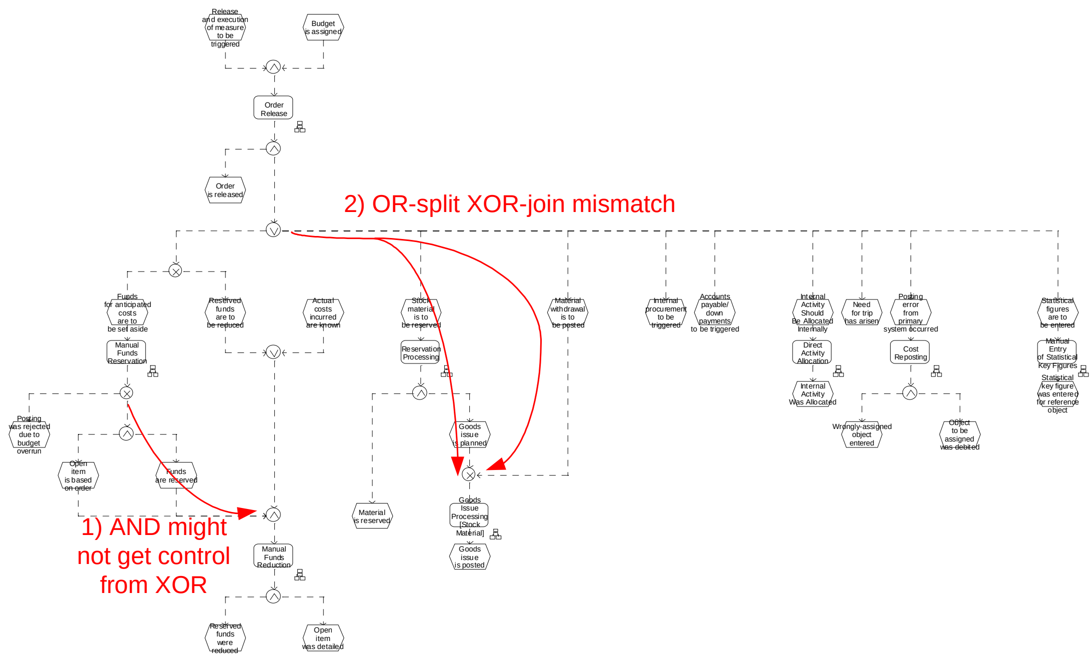


Figure A.5: Asset Accounting – Handling Simple Investment Measures – Release and Implementation of Measure (reduced size 8)

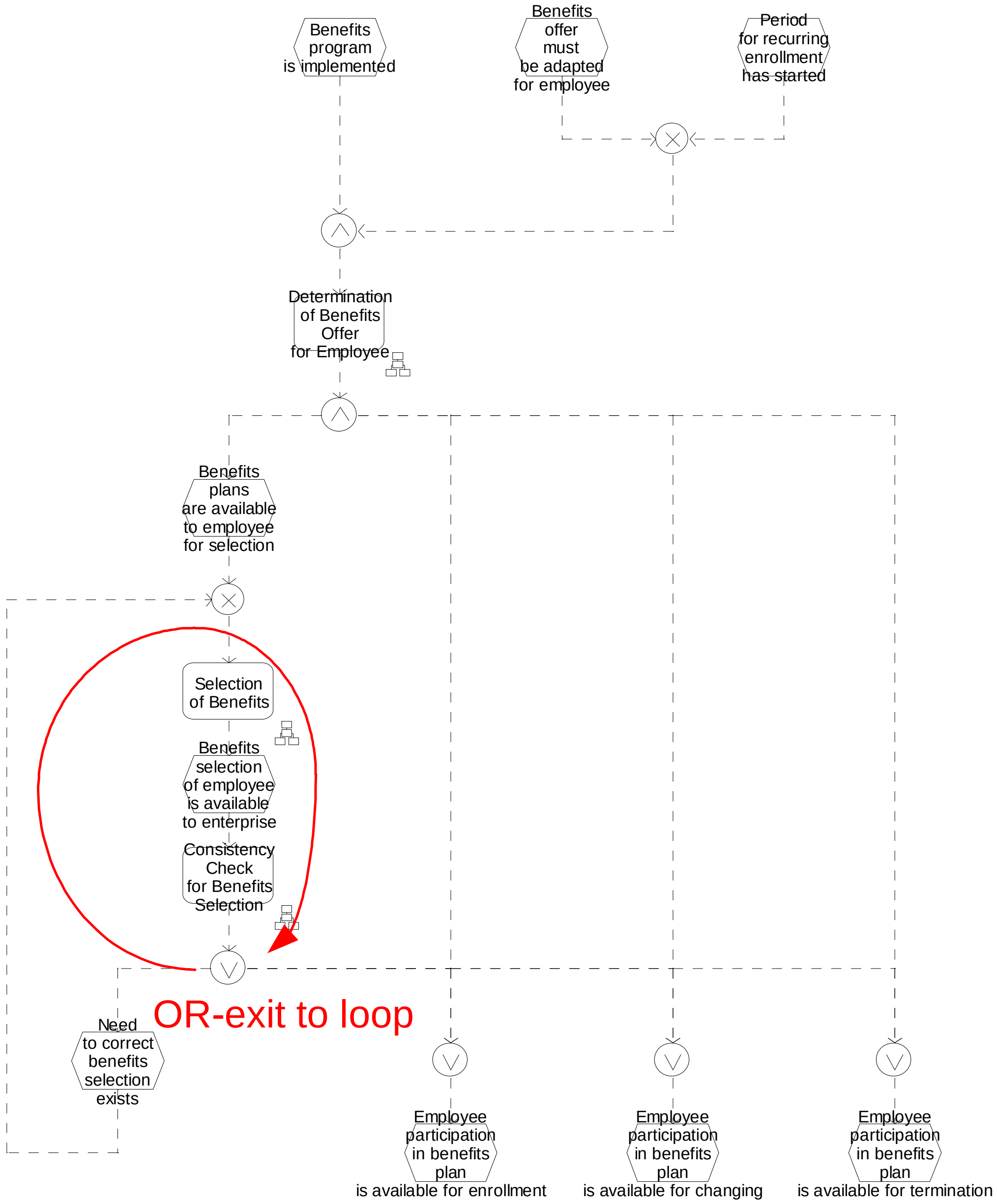


Figure A.6: Benefits Administration – Benefits Administration – Benefits Selection (reduced size 9)

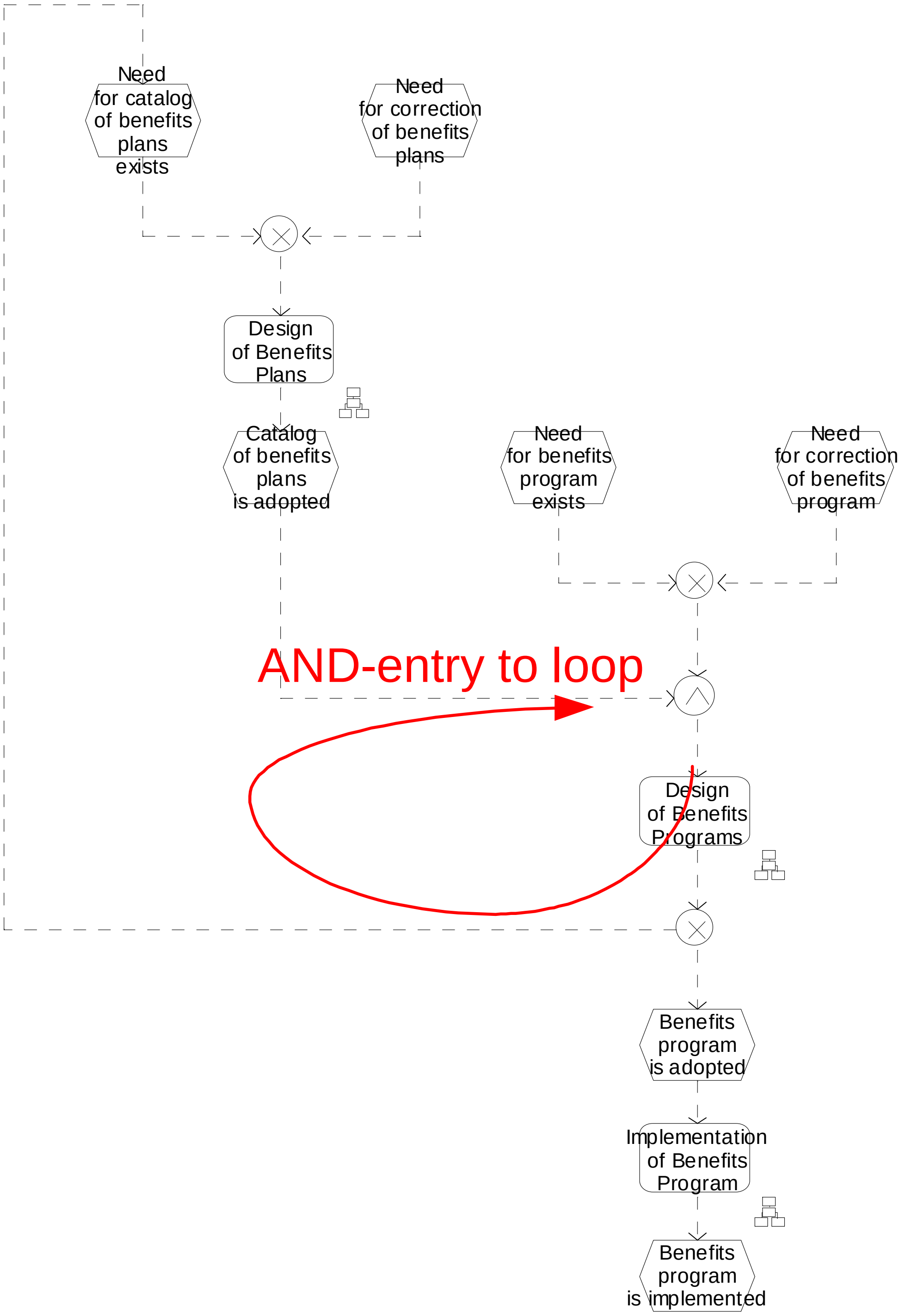


Figure A.7: Benefits Administration – Benefits Administration – Design of Enterprise Benefits System (completely reduced)

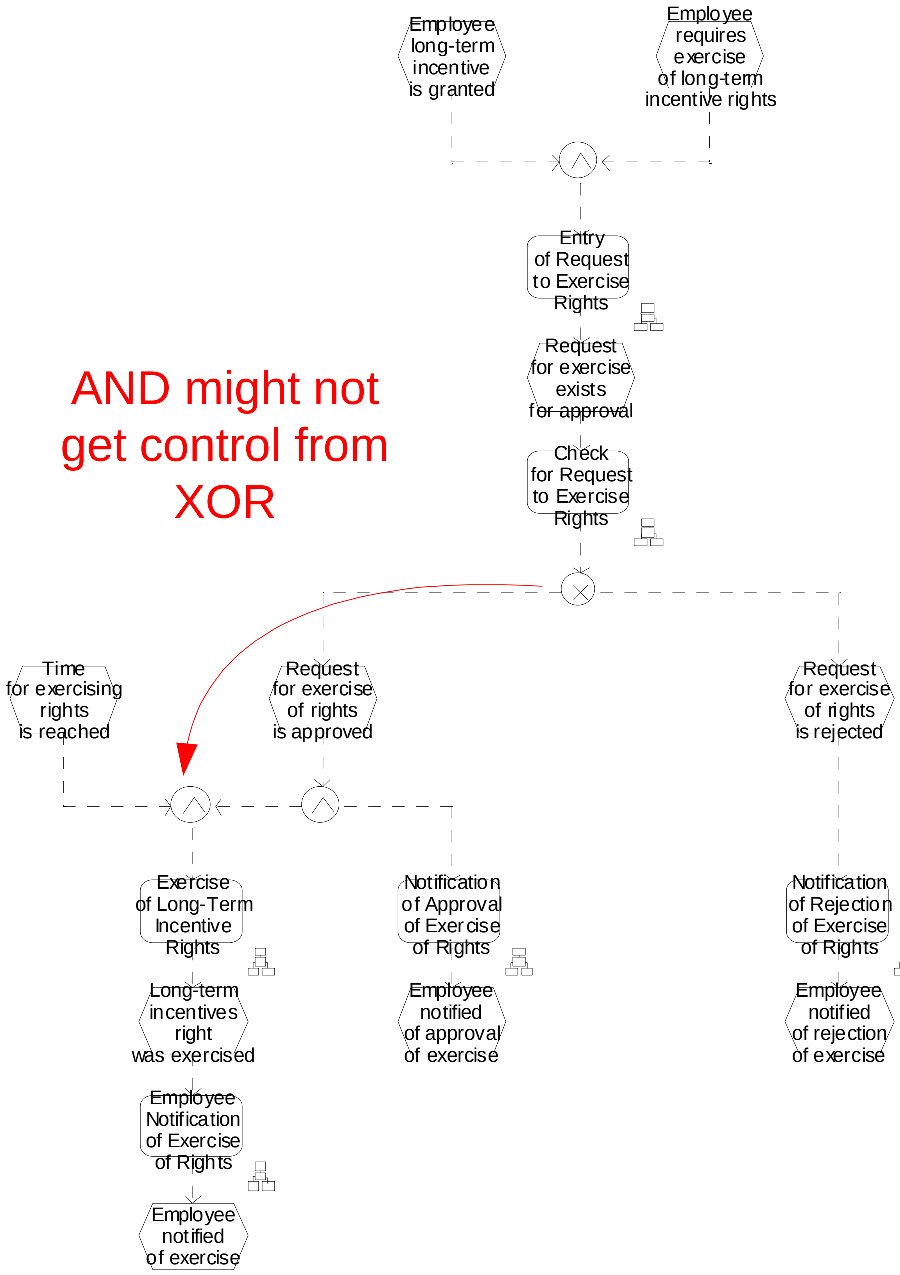


Figure A.8: Compensation Management – Long-Term Incentives – Exercise of Long-Term Incentive Rights by Employee (completely reduced)

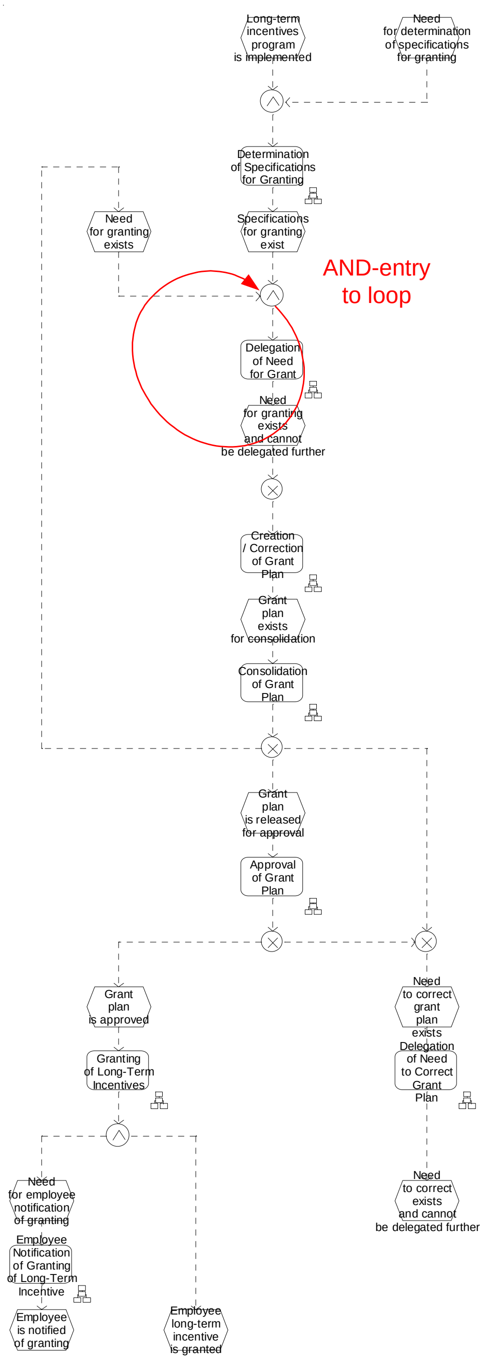


Figure A.9: Compensation Management – Long-Term Incentives – Granting of Share of Long-Term Incentive to Employee (completely reduced)

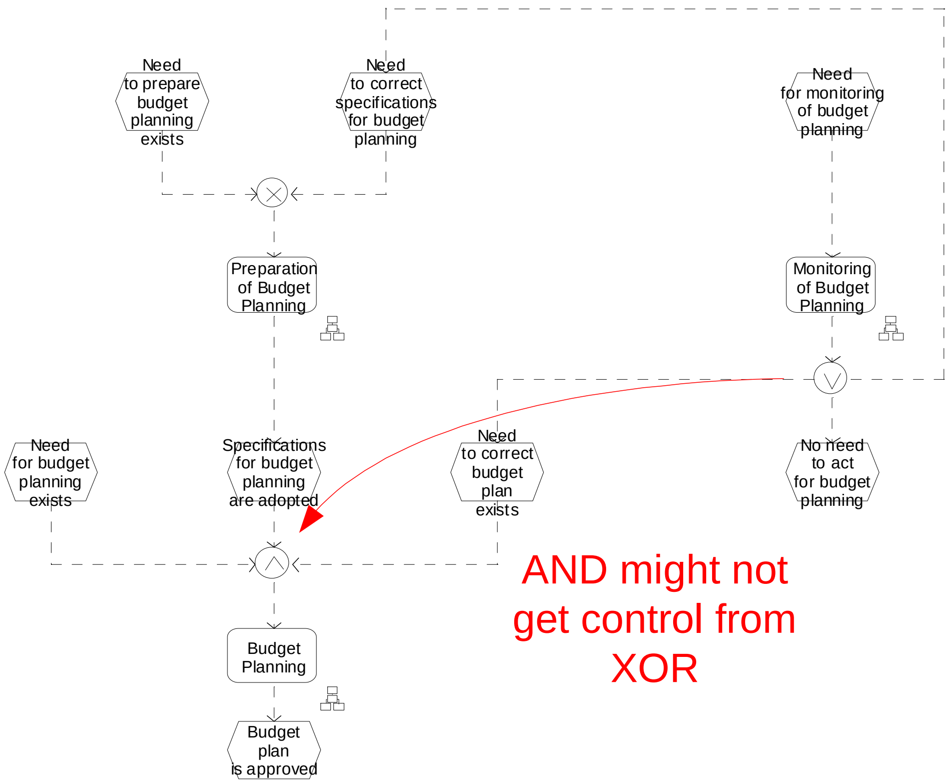


Figure A.10: Compensation Management – Personnel Budget Planning (reduced size 7)

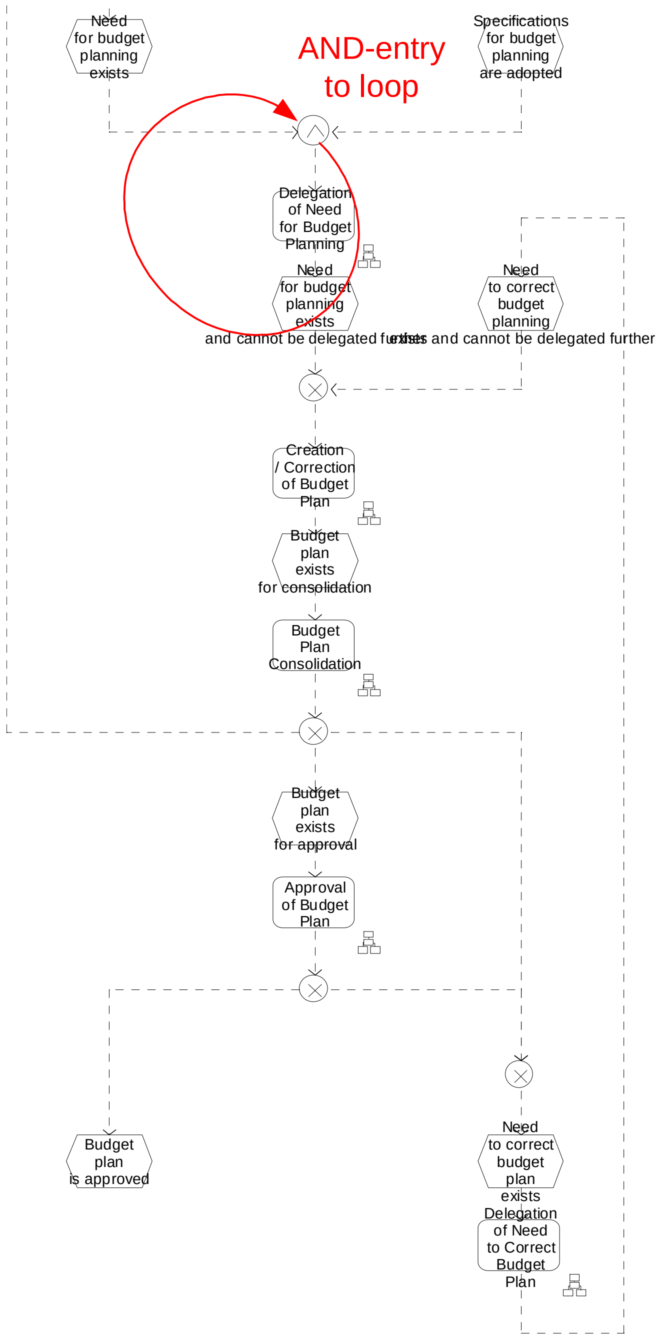


Figure A.11: Compensation Management – Personnel Budget Planning – Budget Planning (completely reduced)

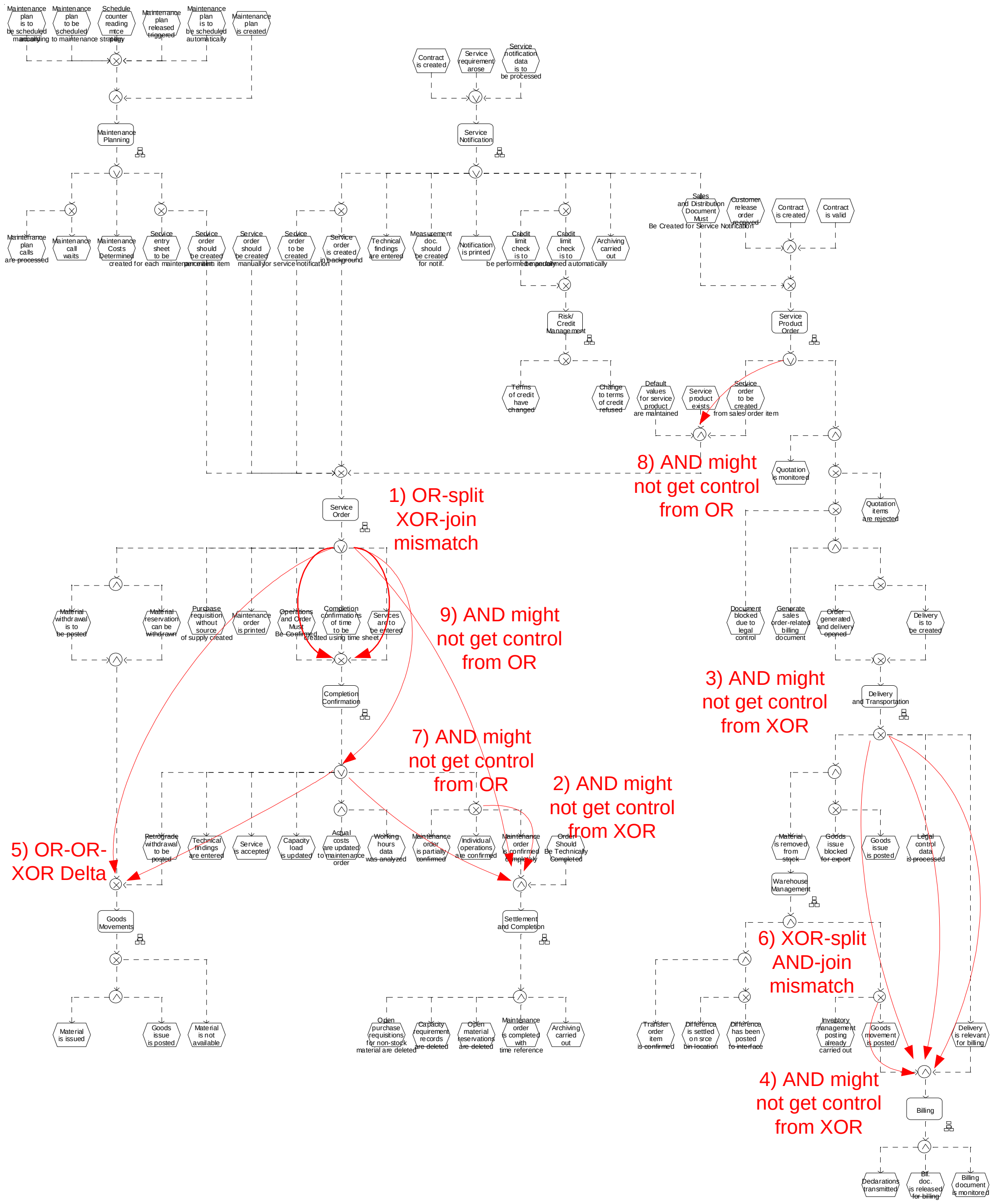


Figure A.12: Customer Service – Repairs Processing at Customer (Field Service) (reduced size 17)

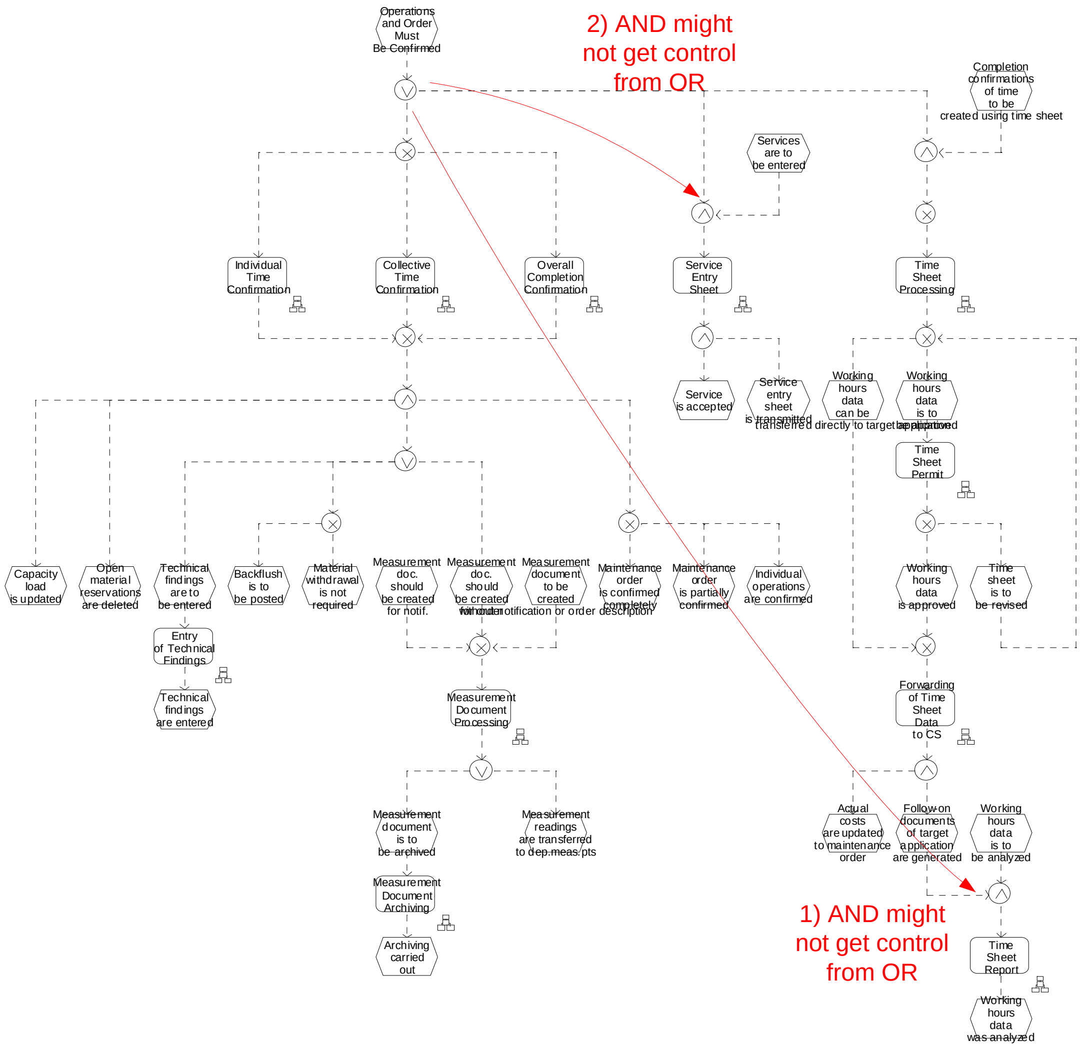


Figure A.13: Customer Service – Repairs Processing at Customer (Field Service) – Completion Confirmation (reduced size 11)

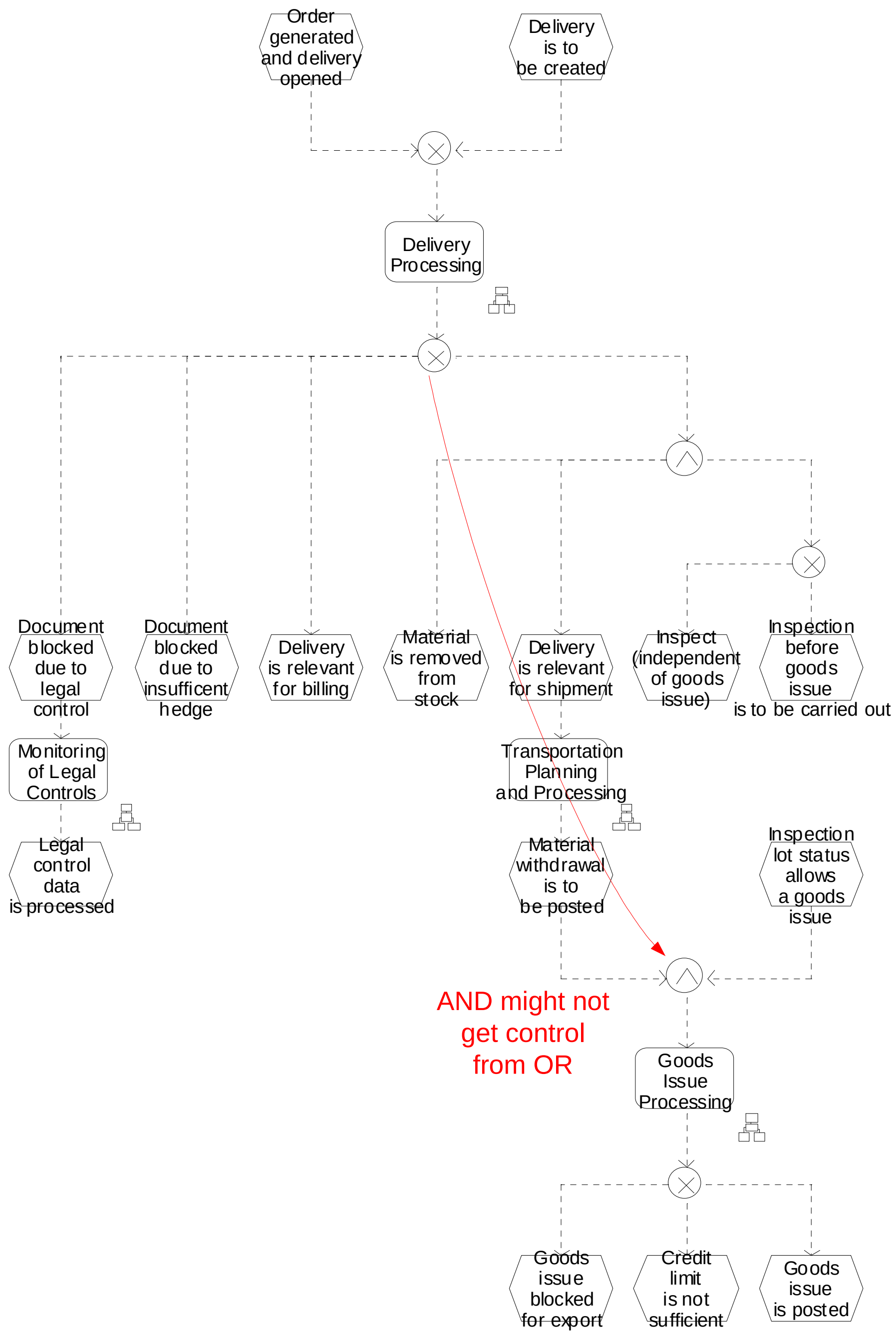


Figure A.14: Customer Service – Repairs Processing at Customer (Field Service) – Delivery and Transportation (completely reduced)

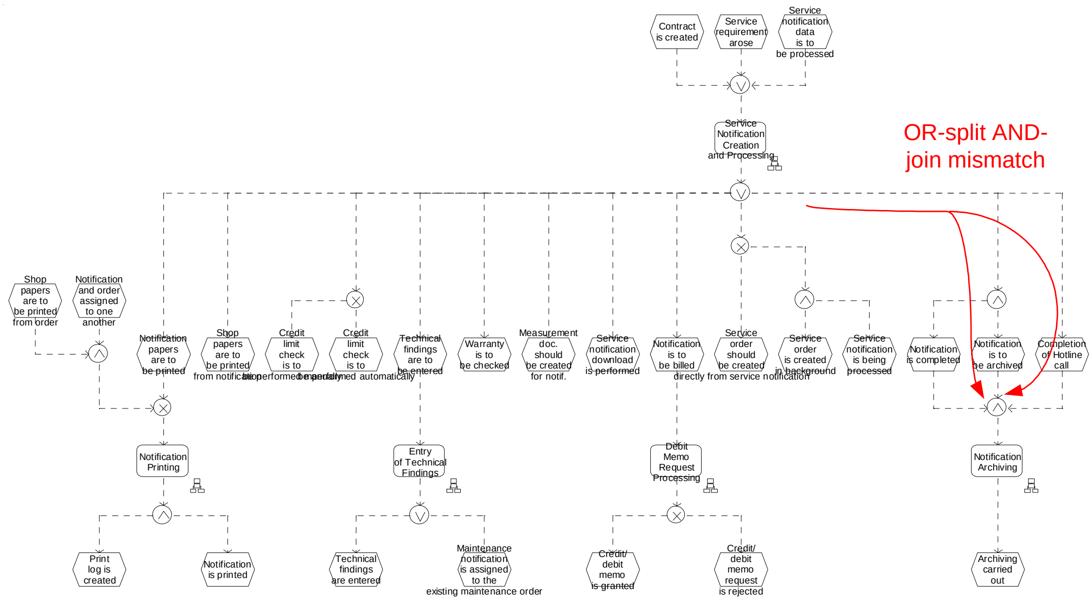


Figure A.15: Customer Service – Repairs Processing at Customer (Field Service) – Service Notification (reduced size 6)

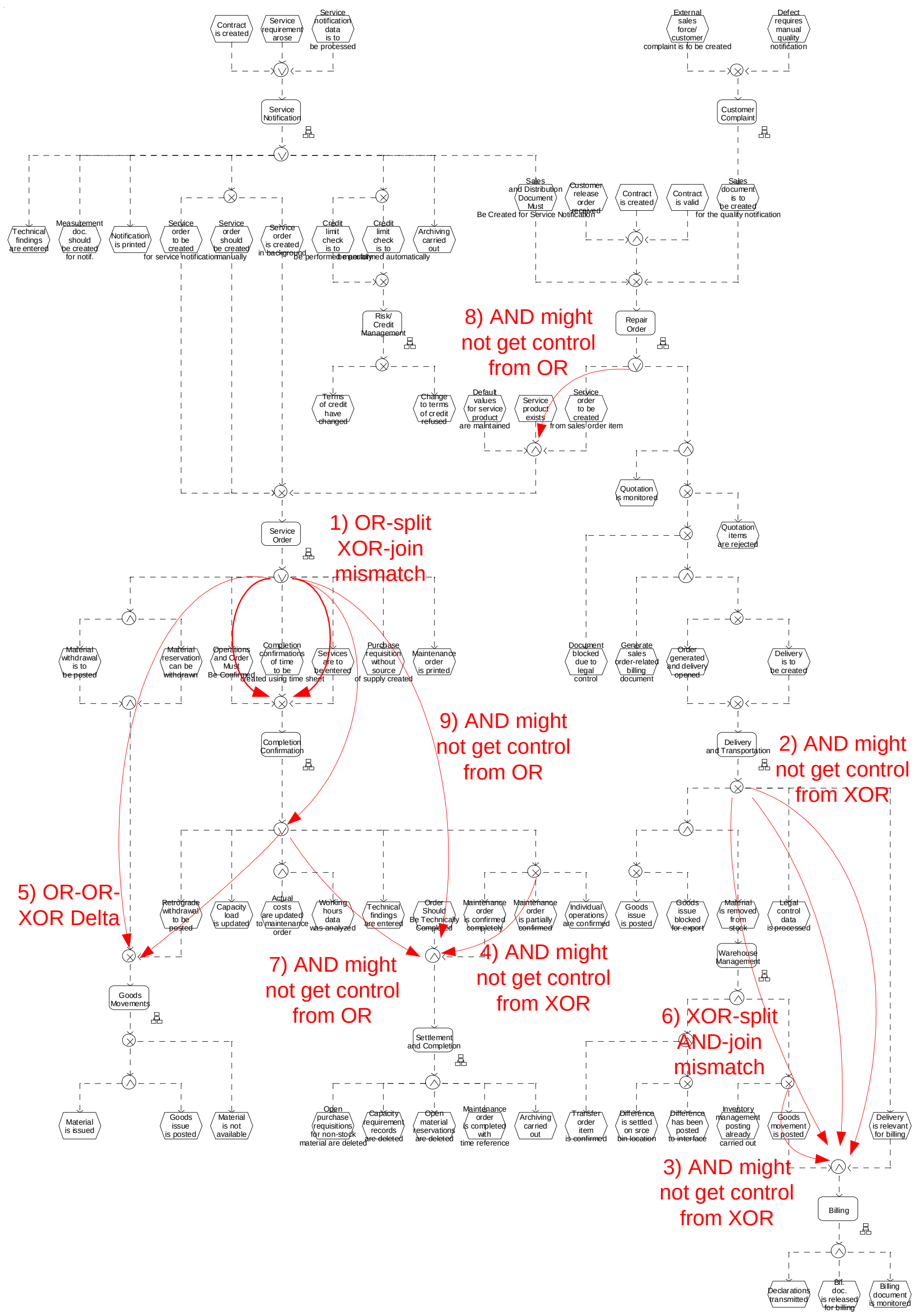


Figure A.16: Customer Service – Repairs Processing in Service Center (Inhouse) (reduced size 12)

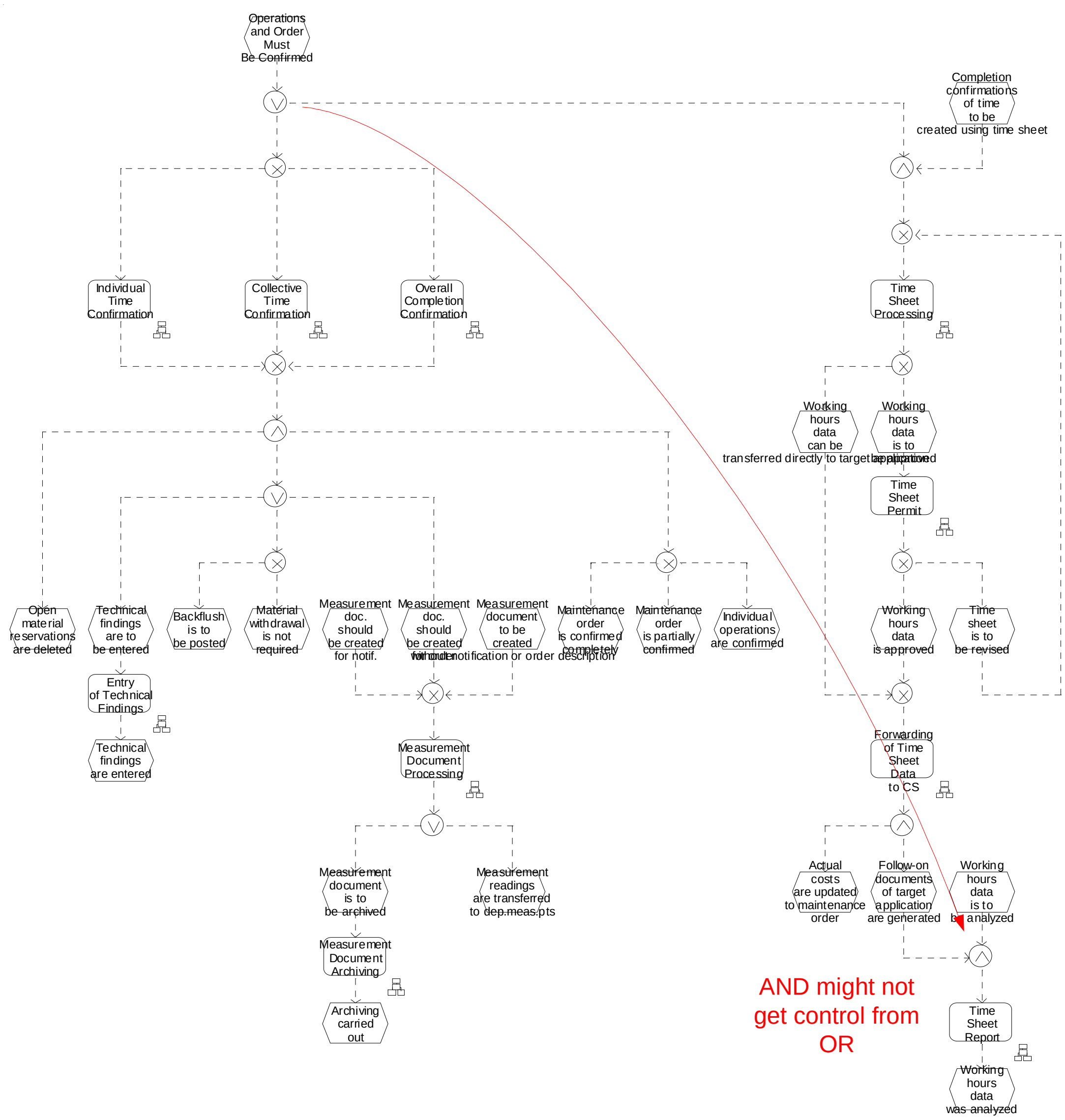


Figure A.17: Customer Service – Repairs Processing in Service Center (Inhouse) – Completion Confirmation (reduced size 8)

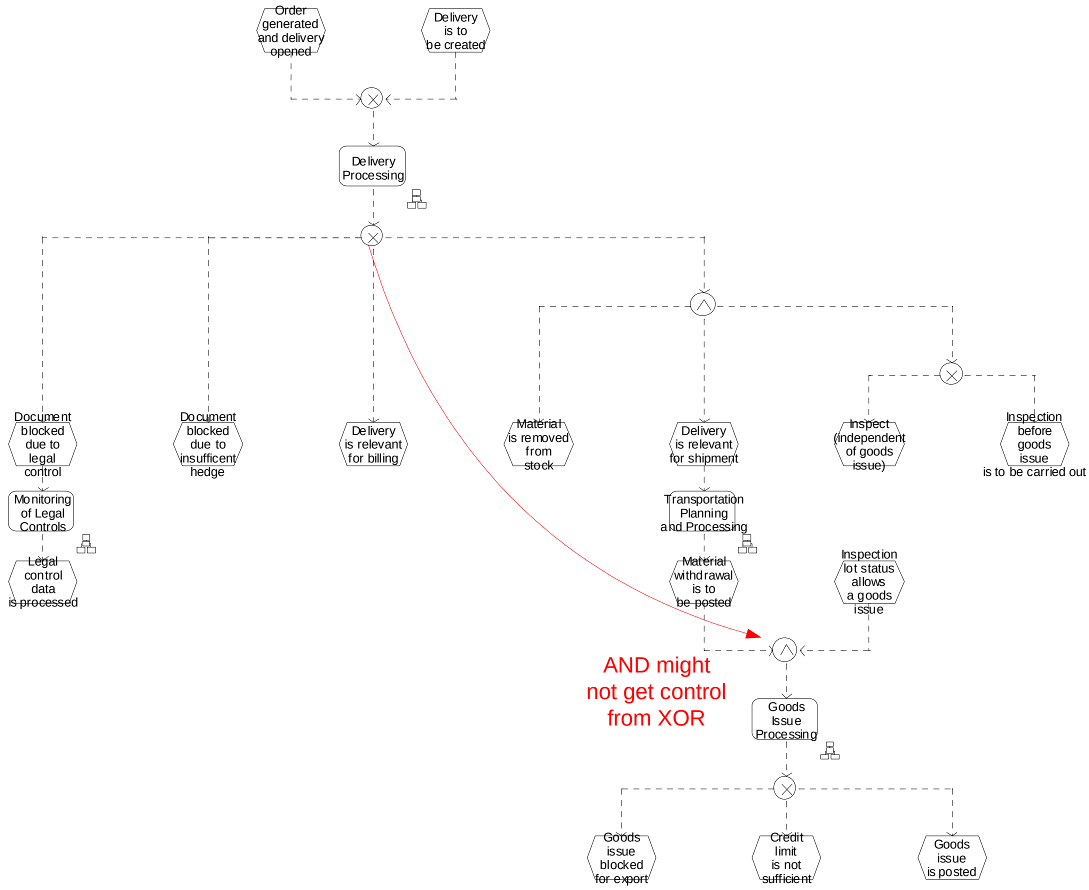


Figure A.18: Customer Service – Spare Parts Delivery Processing – Delivery and Transportation (completely reduced)

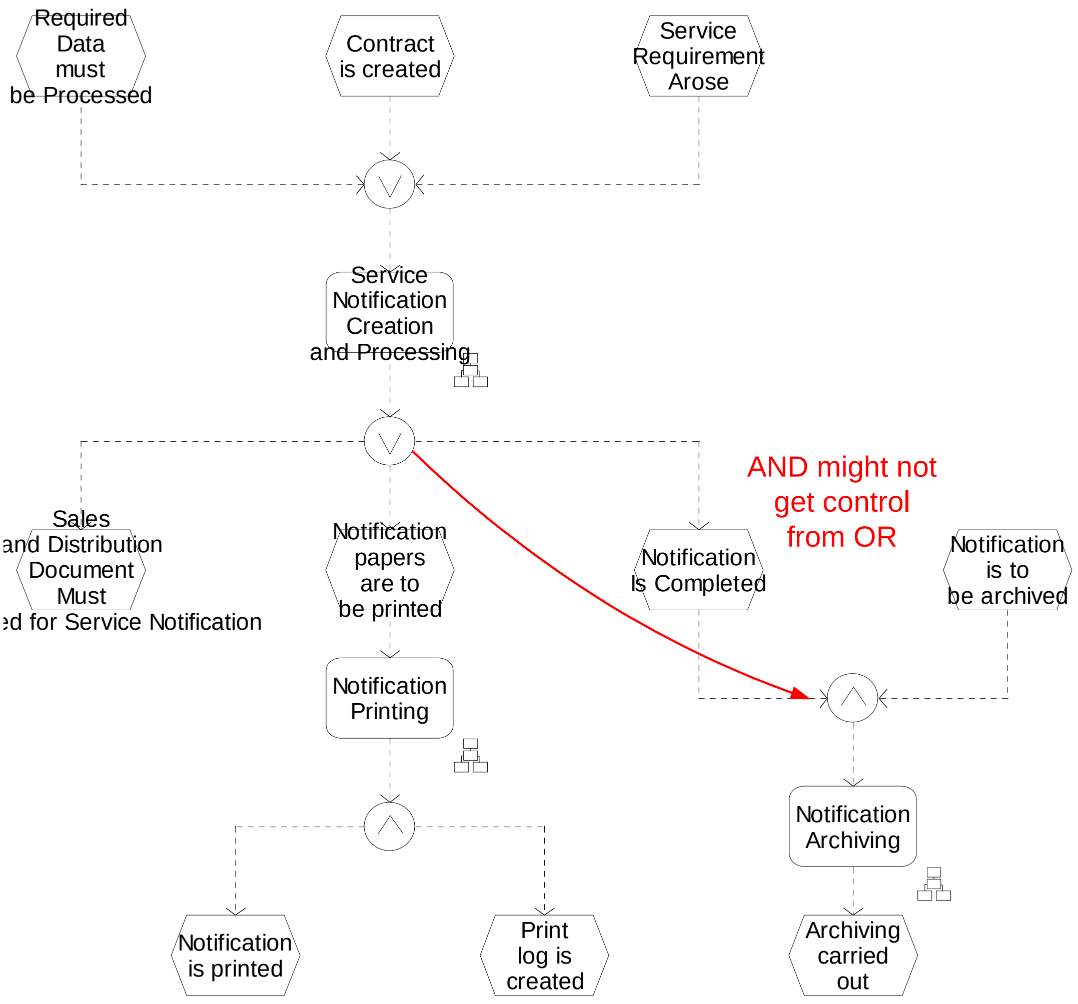


Figure A.19: Customer Service – Spare Parts Delivery Processing – Service Notification (completely reduced)

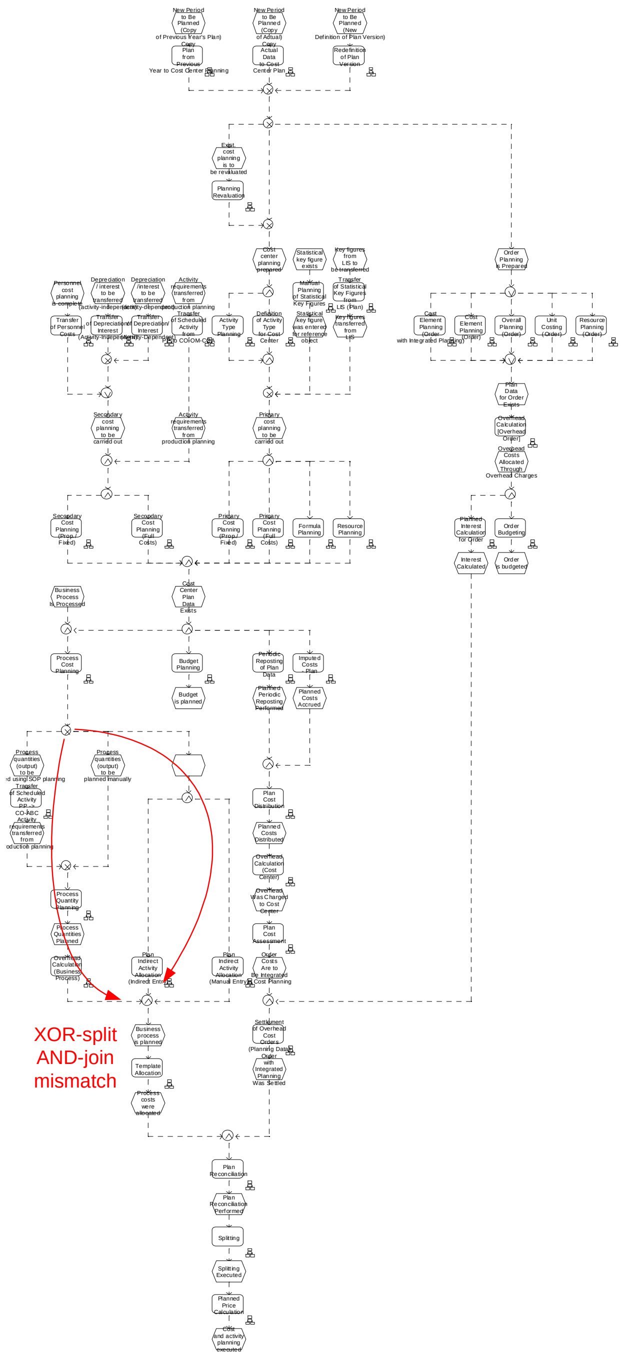


Figure A.20: Enterprise Controlling – Operational business planning – Cost and Activity Planning (reduced size 7)

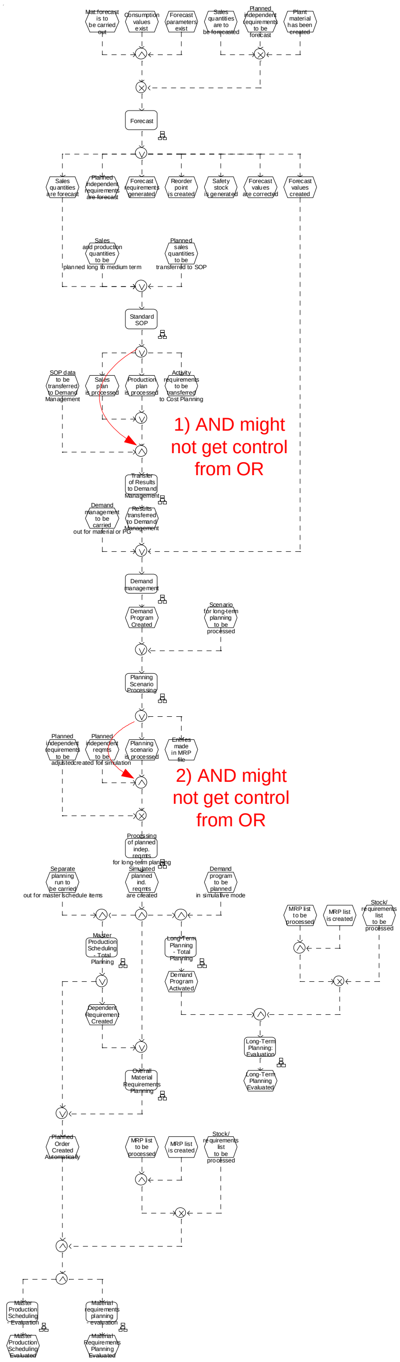


Figure A.21: Enterprise Controlling – Operational business planning – Production Planning (reduced size 23)

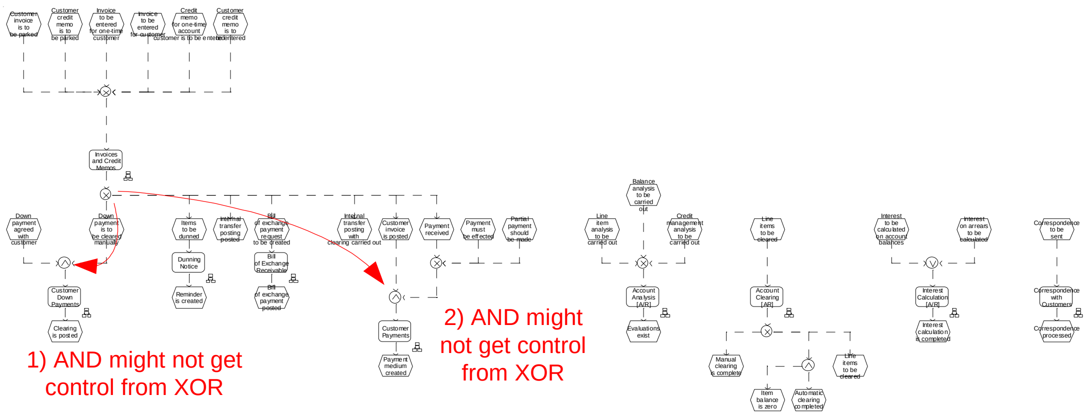


Figure A.22: Financial Accounting – Accounts Receivable (reduced size 9)

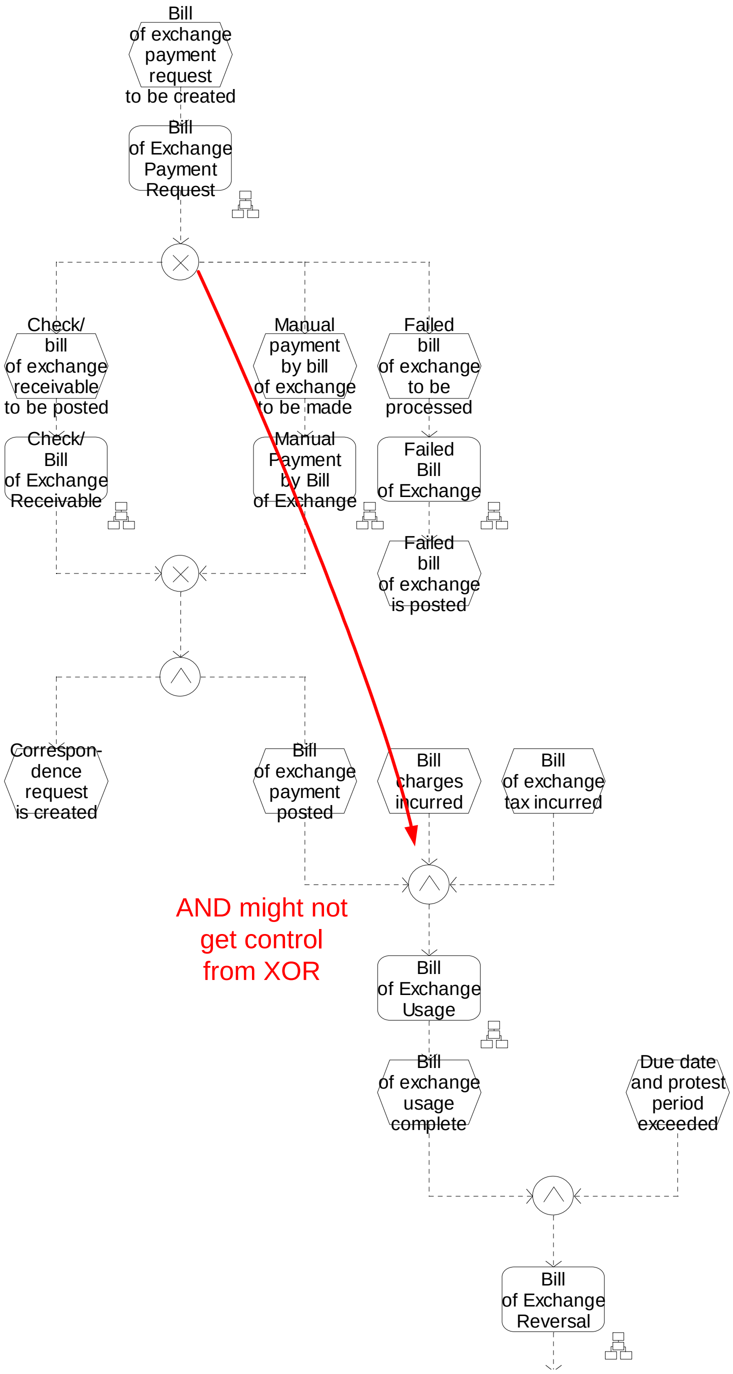


Figure A.23: Financial Accounting – Accounts Receivable – Bill of Exchange Receivable (completely reduced)

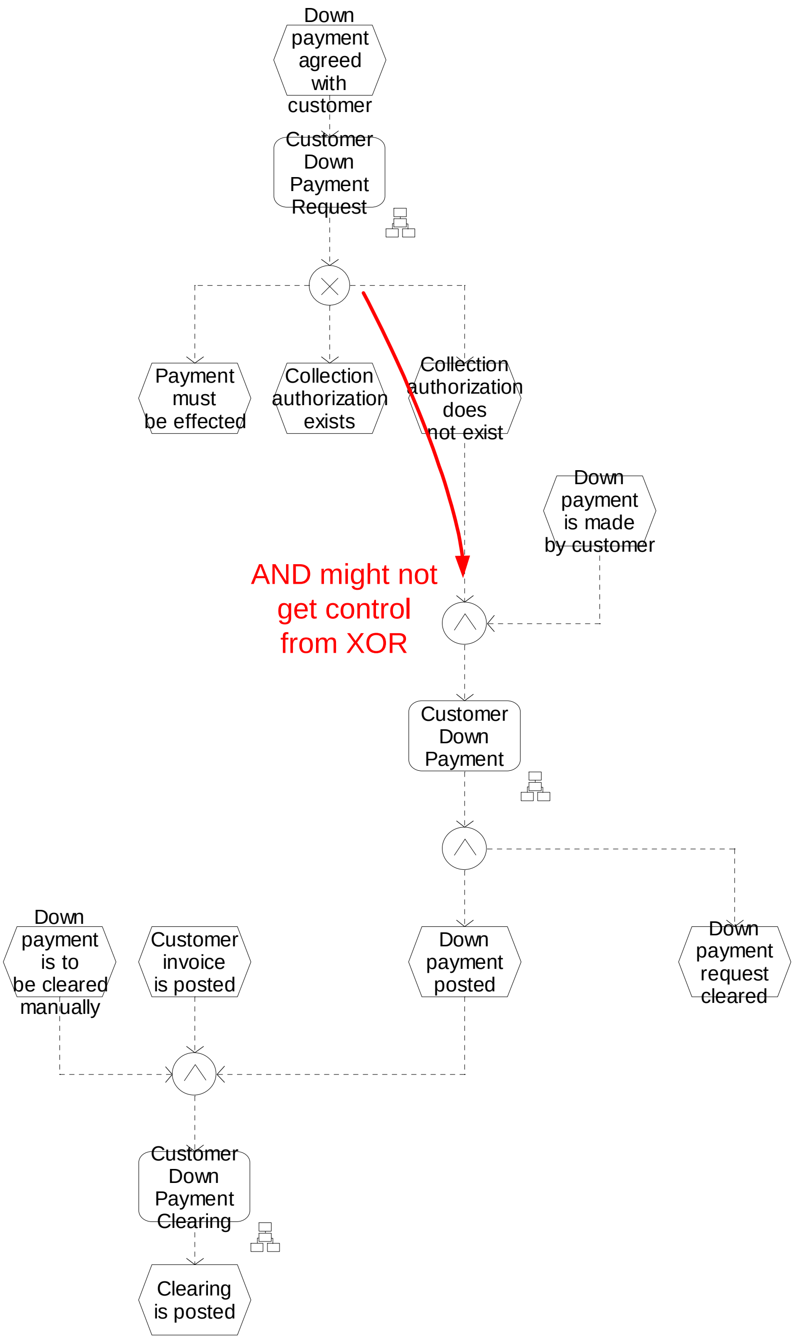


Figure A.24: Financial Accounting – Accounts Receivable – Customer Down Payments (completely reduced)

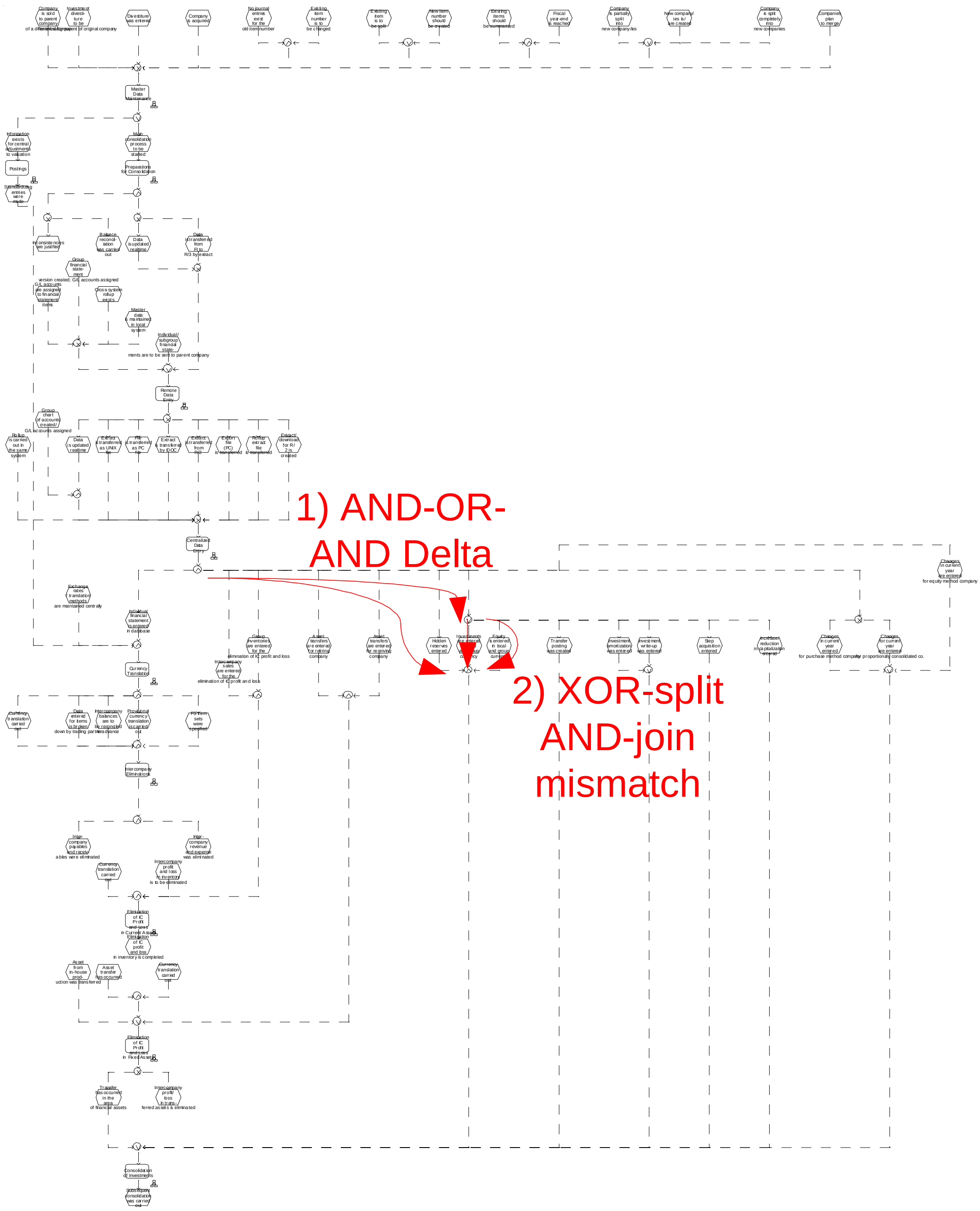


Figure A.25: Financial Accounting – Consolidation (reduced size 22)

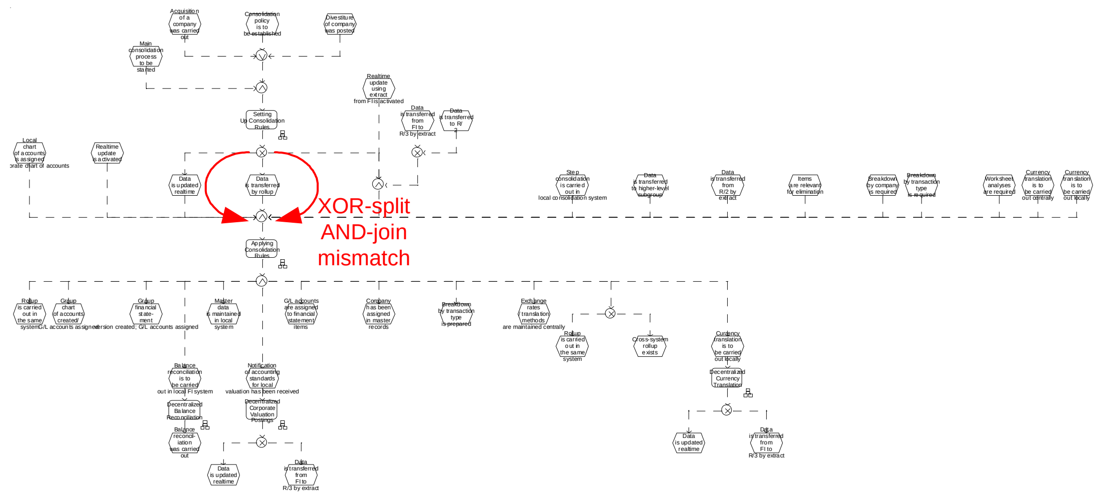


Figure A.26: Financial Accounting – Consolidation – Preparations for Consolidation (completely reduced)

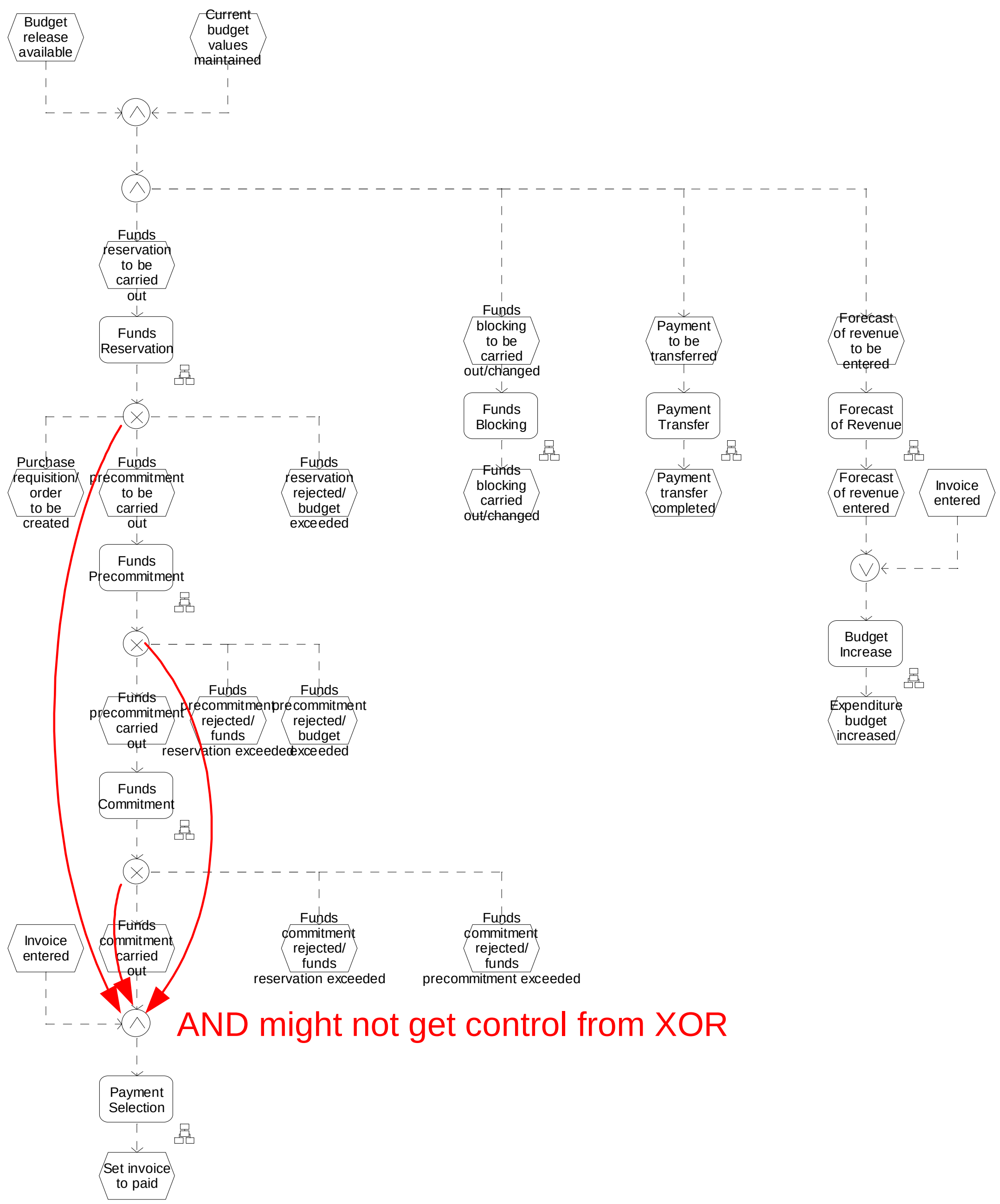


Figure A.27: Financial Accounting – Funds Management – Budget Execution (completely reduced)

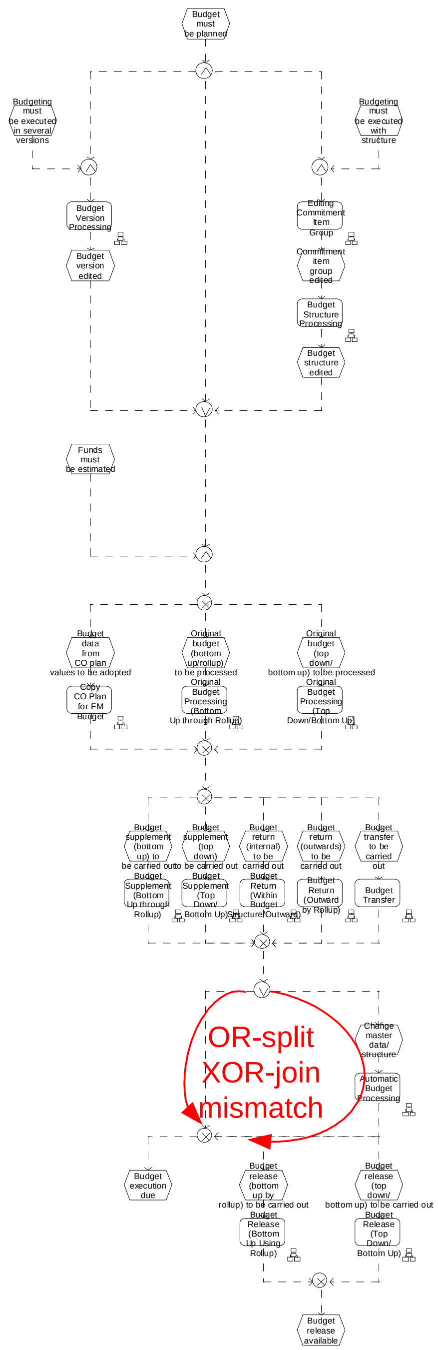


Figure A.28: Financial Accounting – Funds Management – Budget Planning (completely reduced)

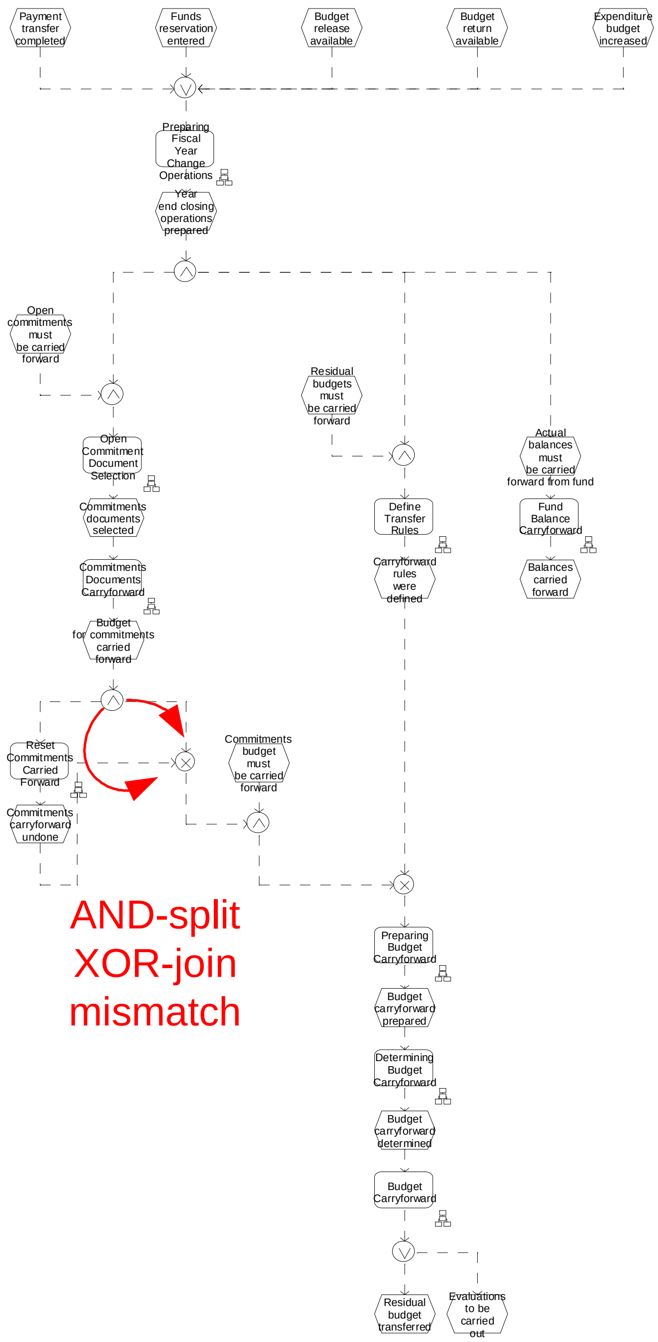


Figure A.29: Financial Accounting – Funds Management – Fiscal Year Change Operations (Funds Management) (reduced size 8)

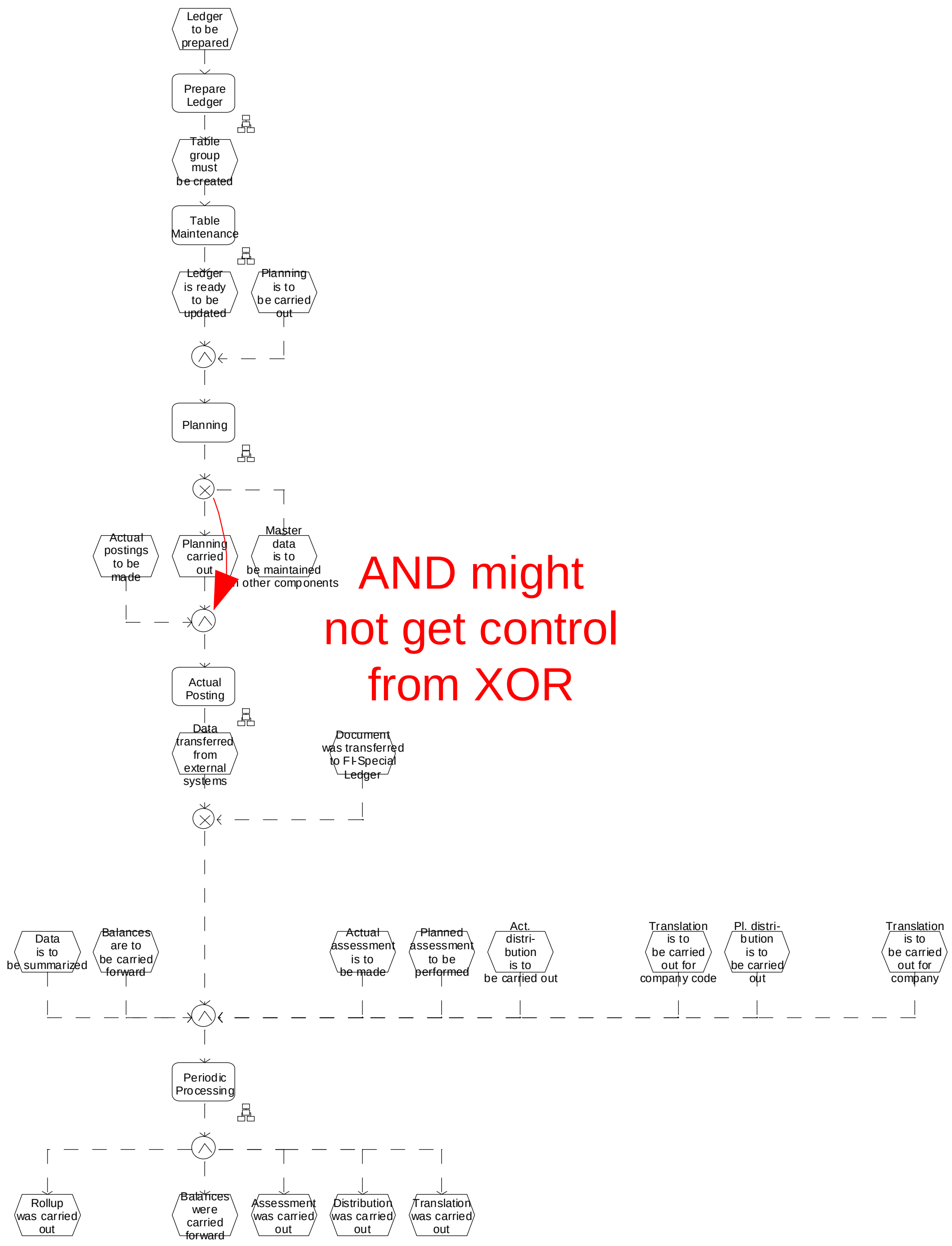


Figure A.30: Financial Accounting – Special Purpose Ledger (completely reduced)

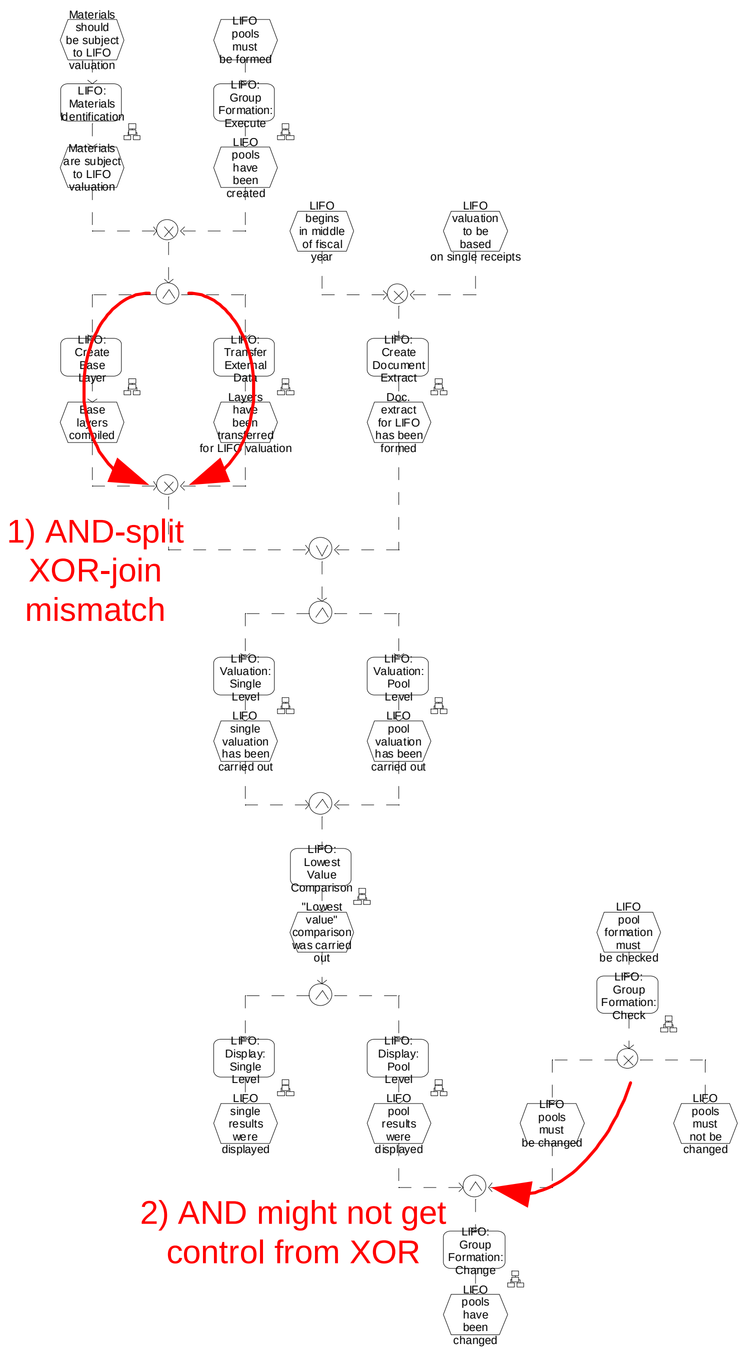


Figure A.31: Financial Accounting – Valuation of Balances Relevant to Balance Sheet – LIFO valuation (completely reduced)

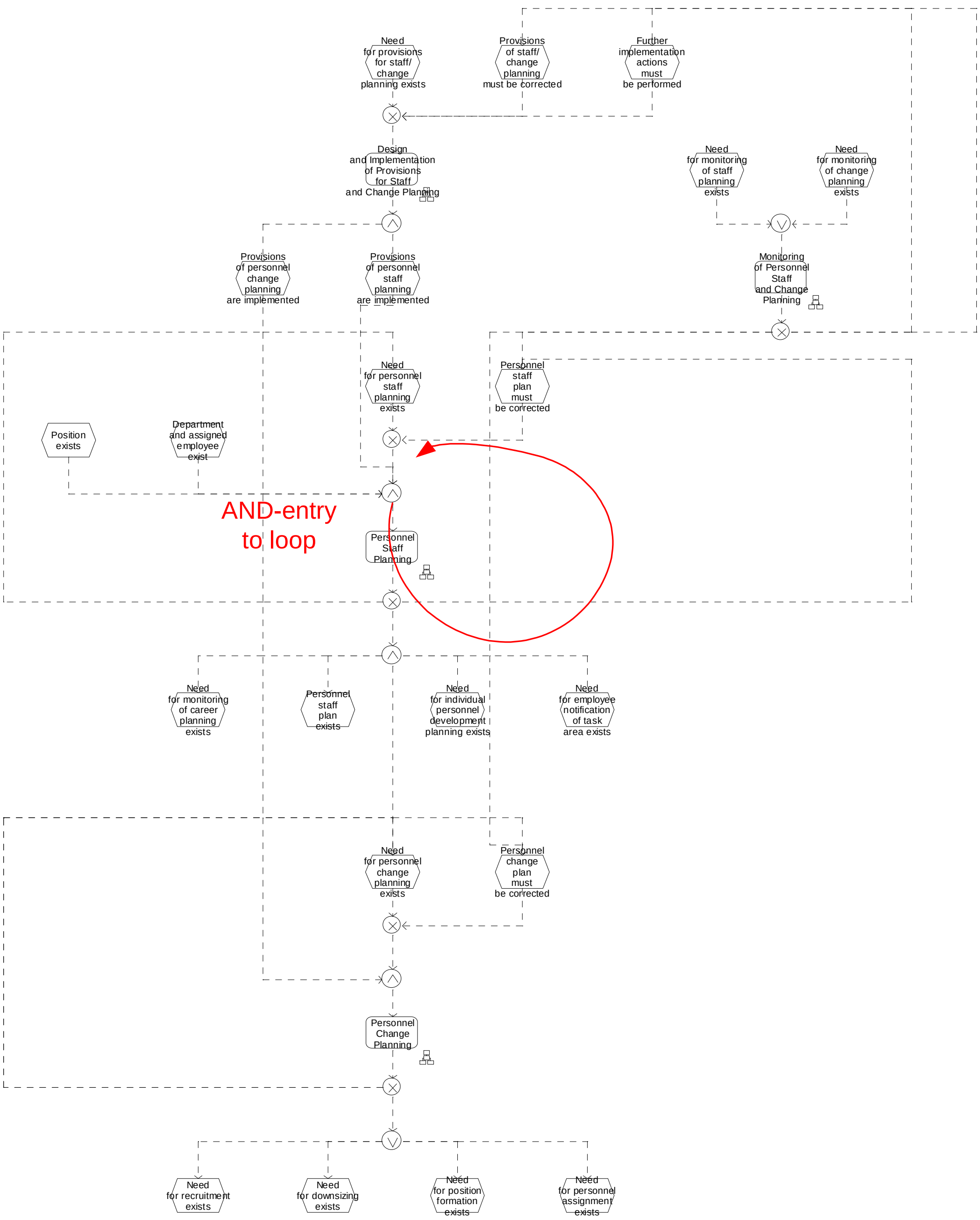


Figure A.32: Organizational Management – Planning Staff Assignment and Changes (reduced size 15)

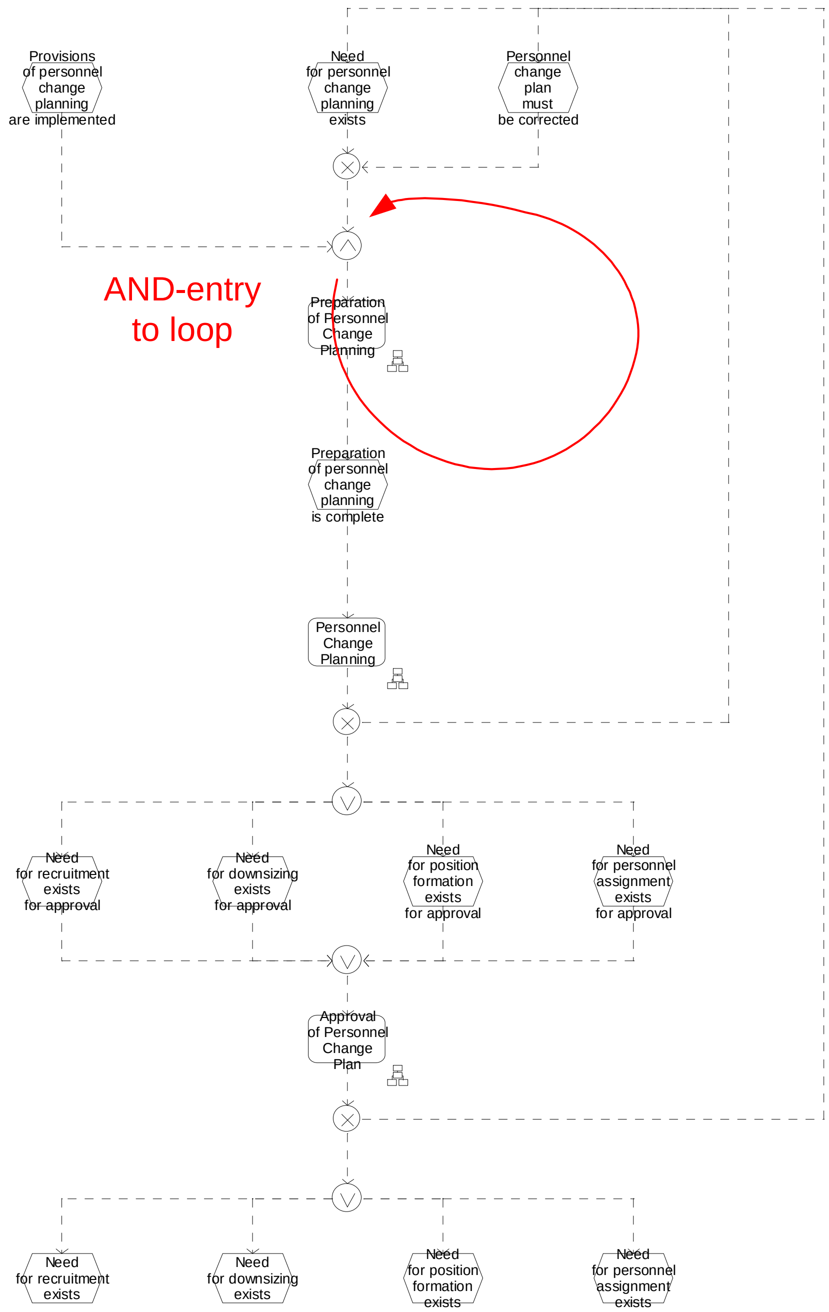


Figure A.33: Organizational Management – Planning Staff Assignment and Changes – Personnel Change Planning (completely reduced)

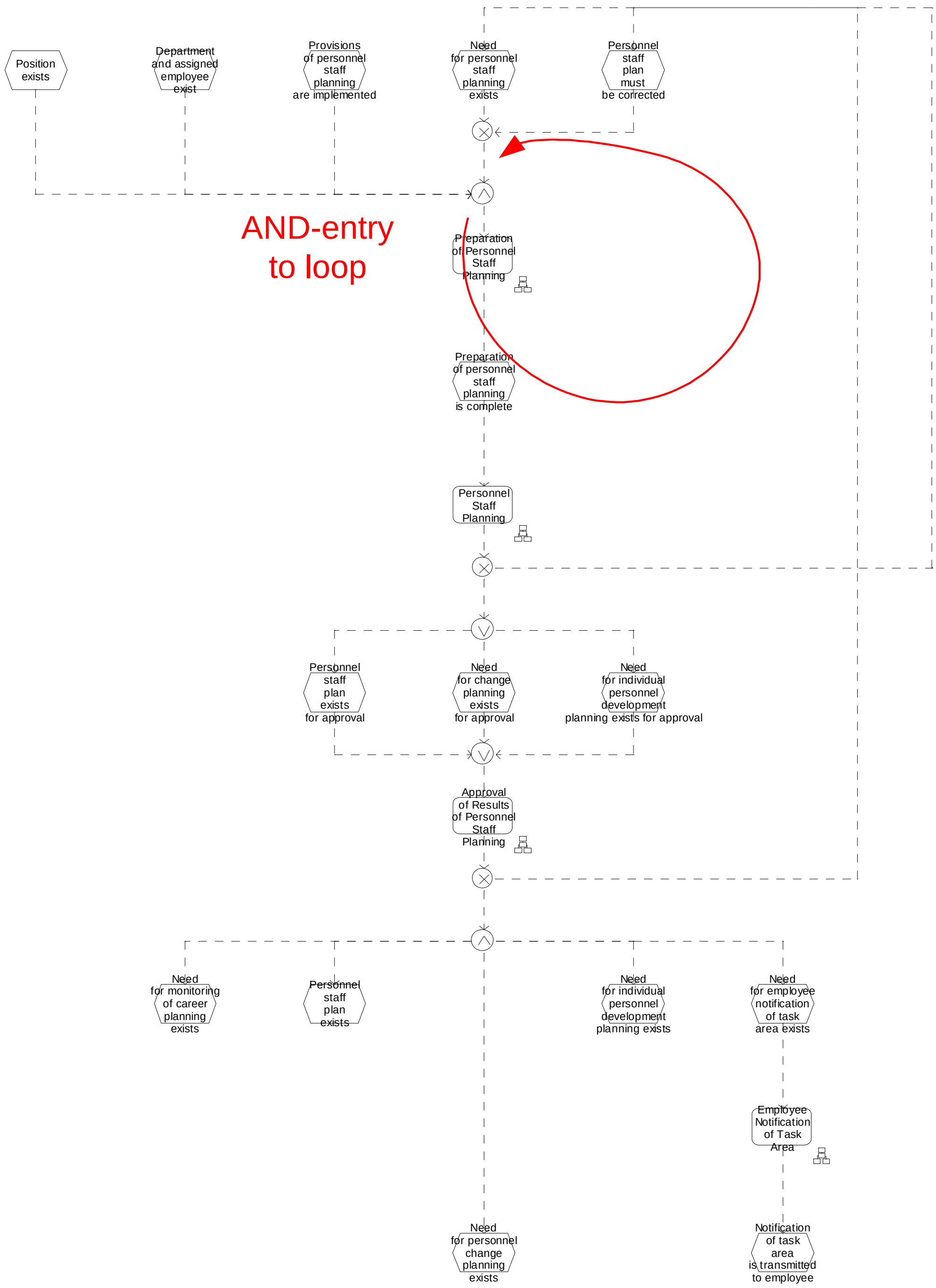


Figure A.34: Organizational Management – Planning Staff Assignment and Changes – Personnel Staff Planning (completely reduced)

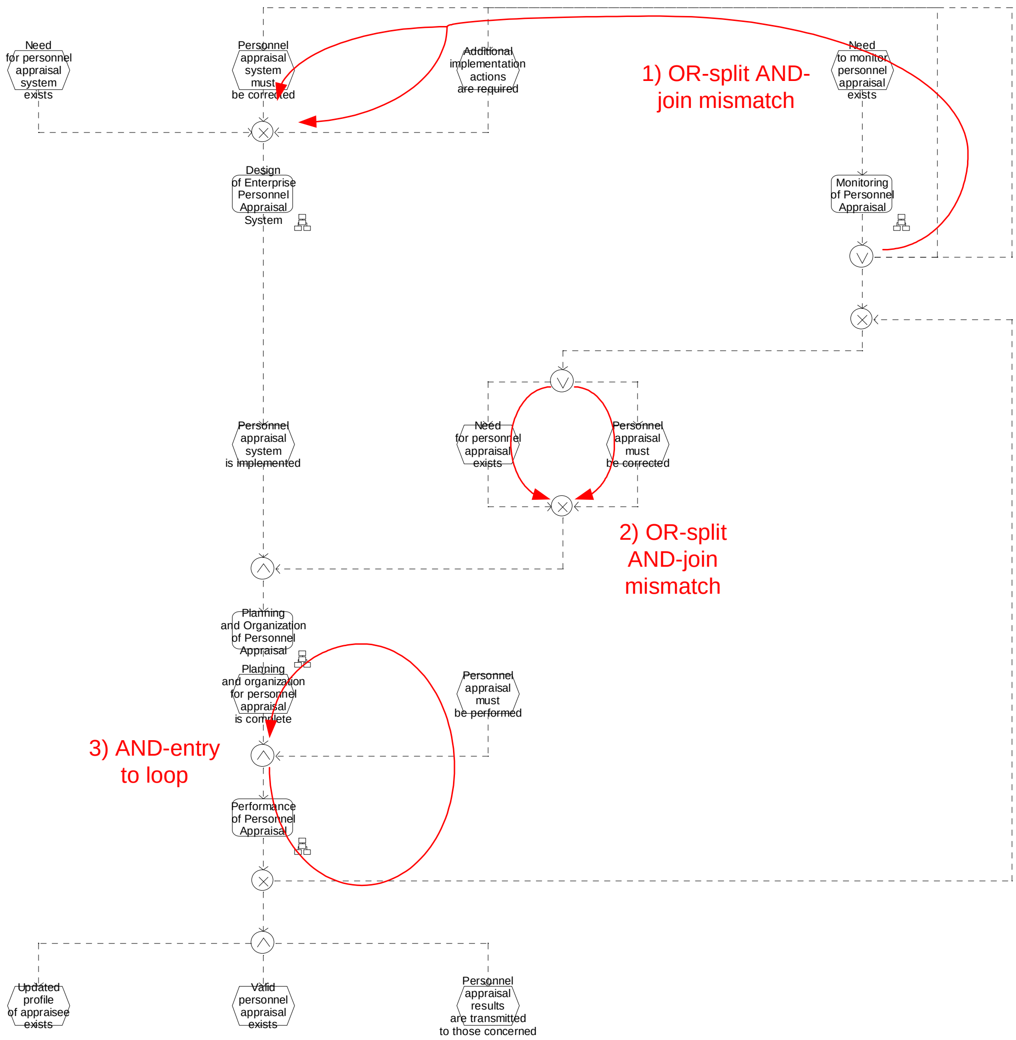


Figure A.35: Personnel Development – Personnel Appraisal (reduced size 8)

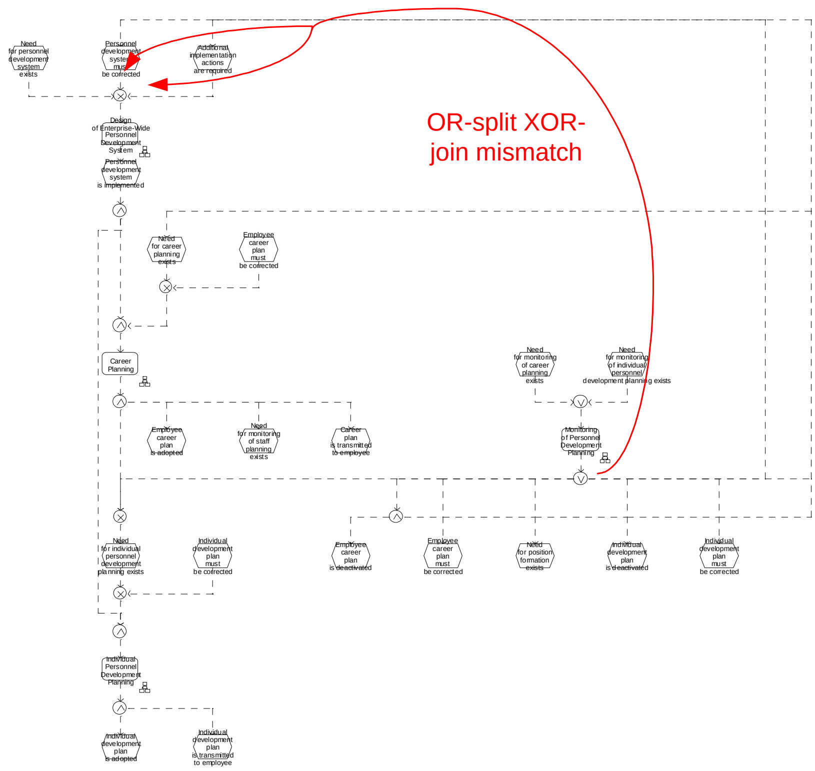


Figure A.36: Personnel Development – Personnel Development Planning (reduced size 13)

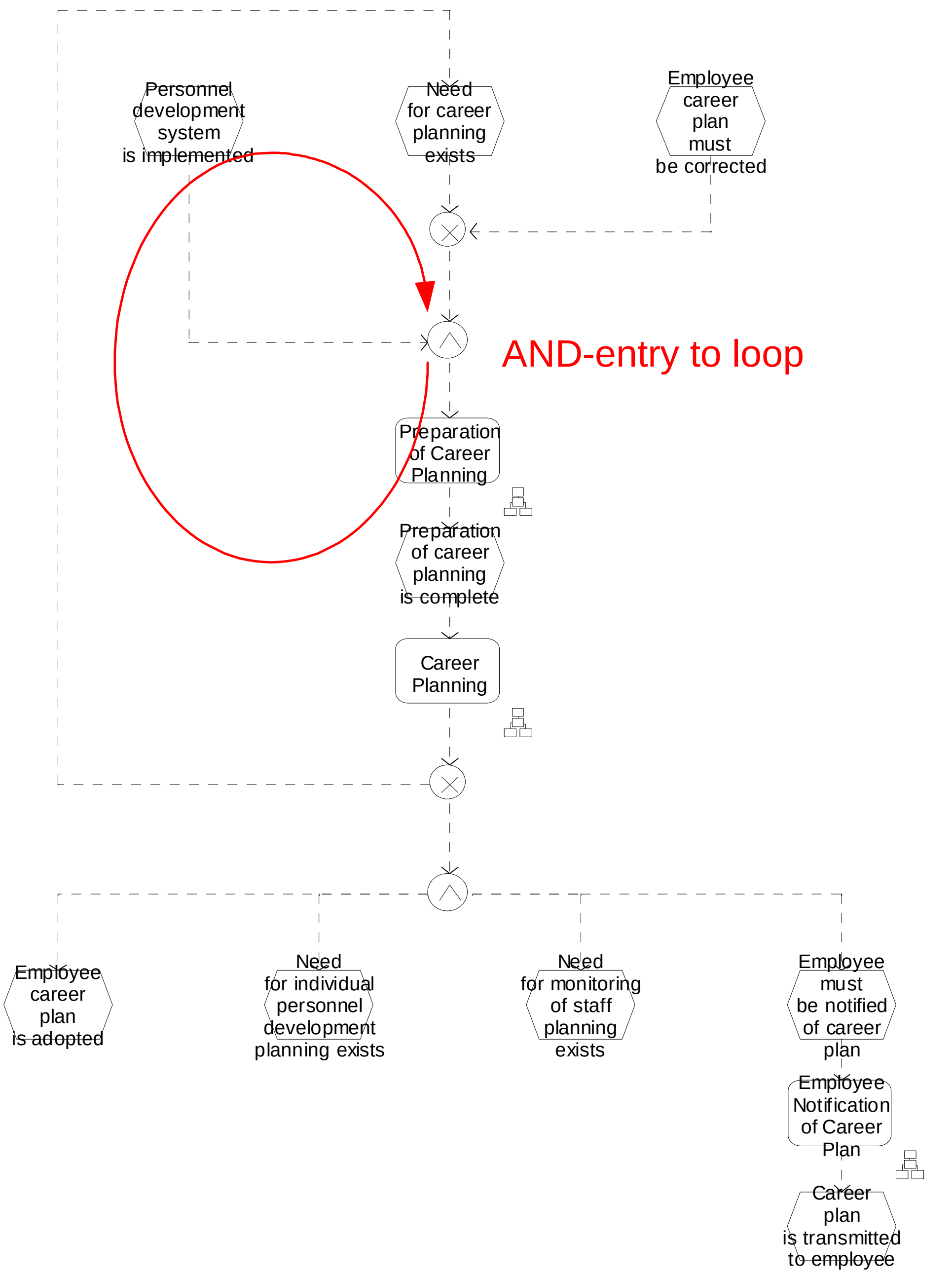


Figure A.37: Personnel Development – Personnel Development Planning – Career Planning (completely reduced)

Figure A.38: Personnel Development – Personnel Development Planning – Individual Personnel Development Planning (completely reduced)

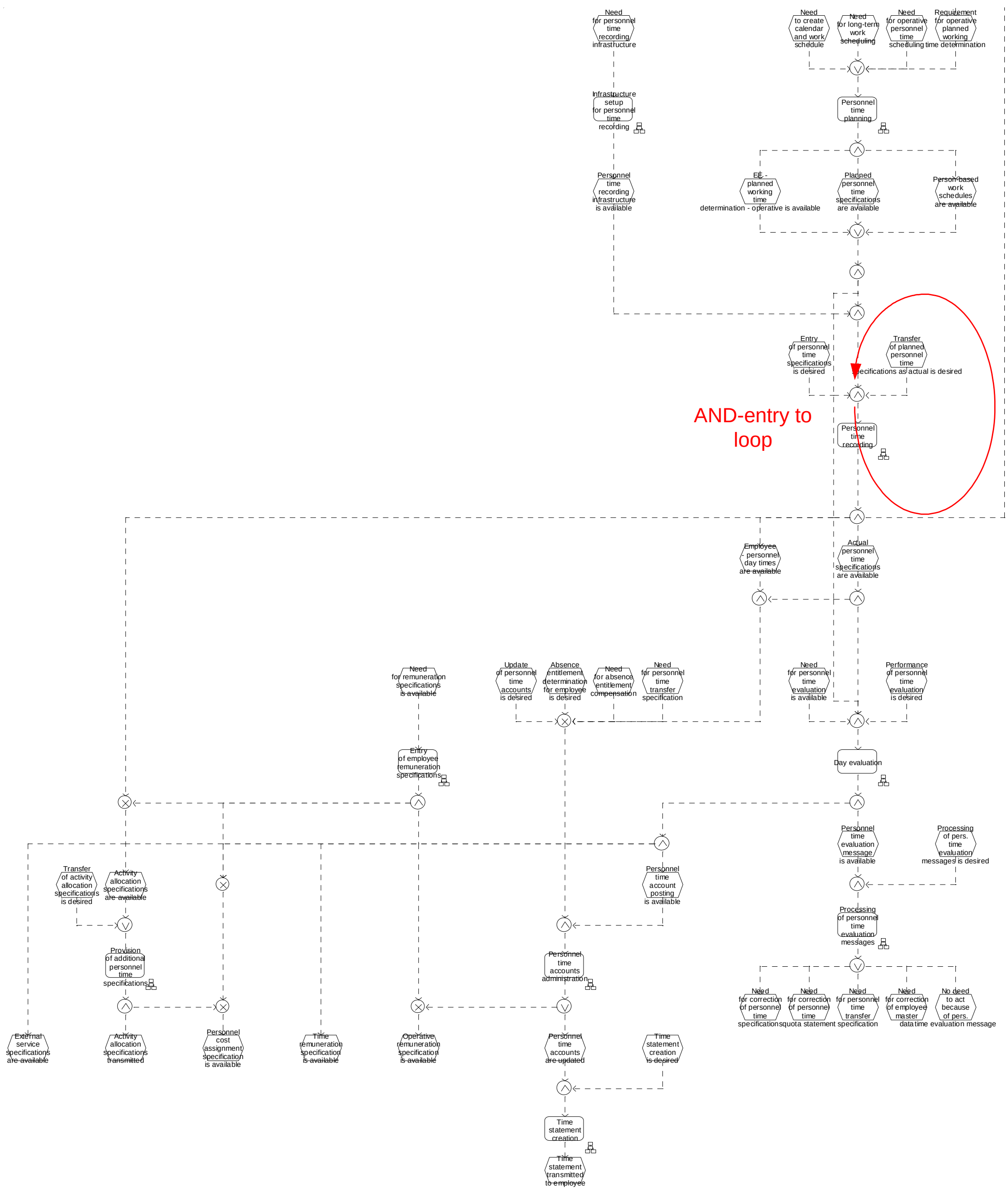


Figure A.39: Personnel Time Management – Personnel Time Management (reduced size 28)

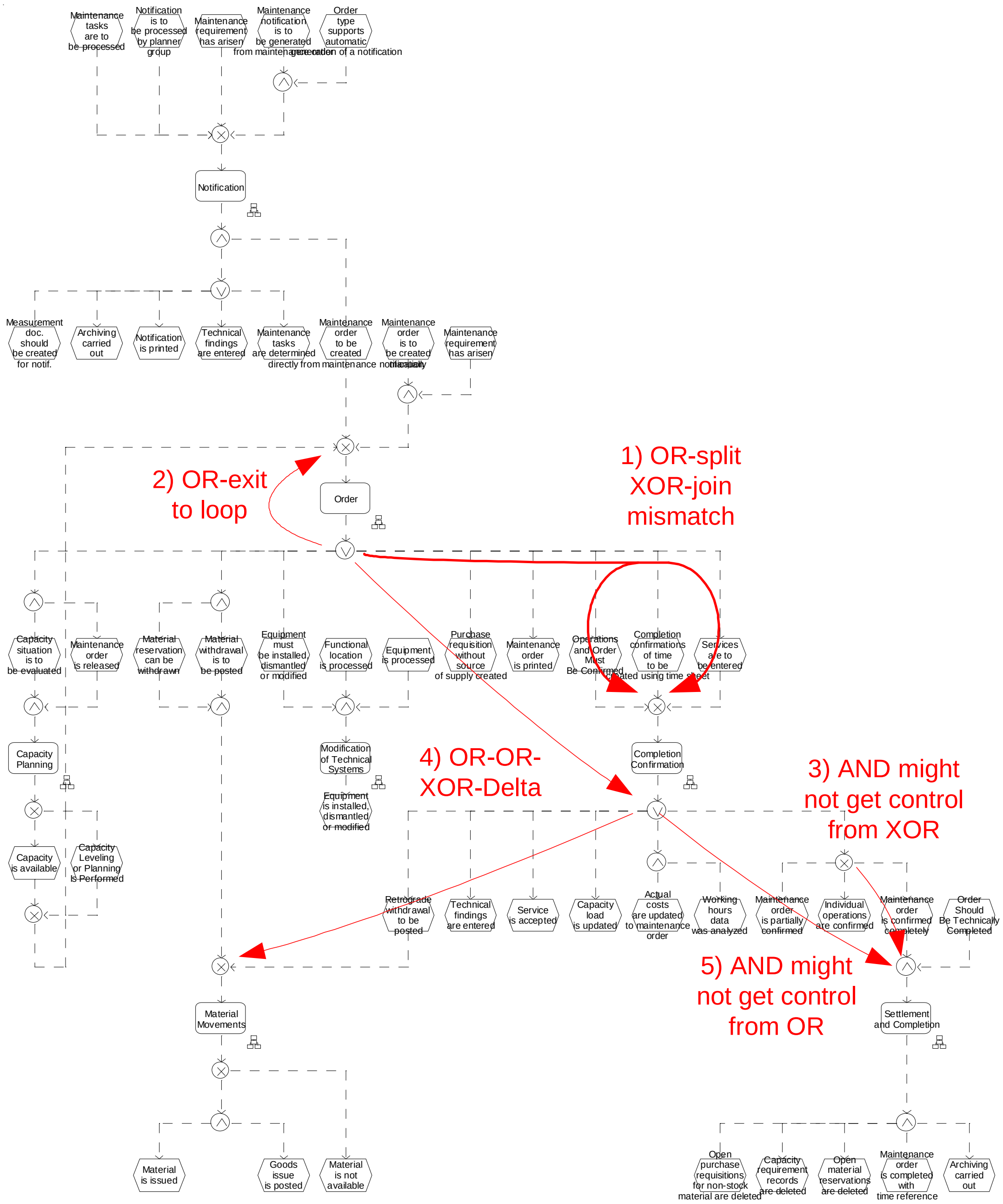


Figure A.40: Plant Maintenance – Breakdown Maintenance Processing (reduced size 9)

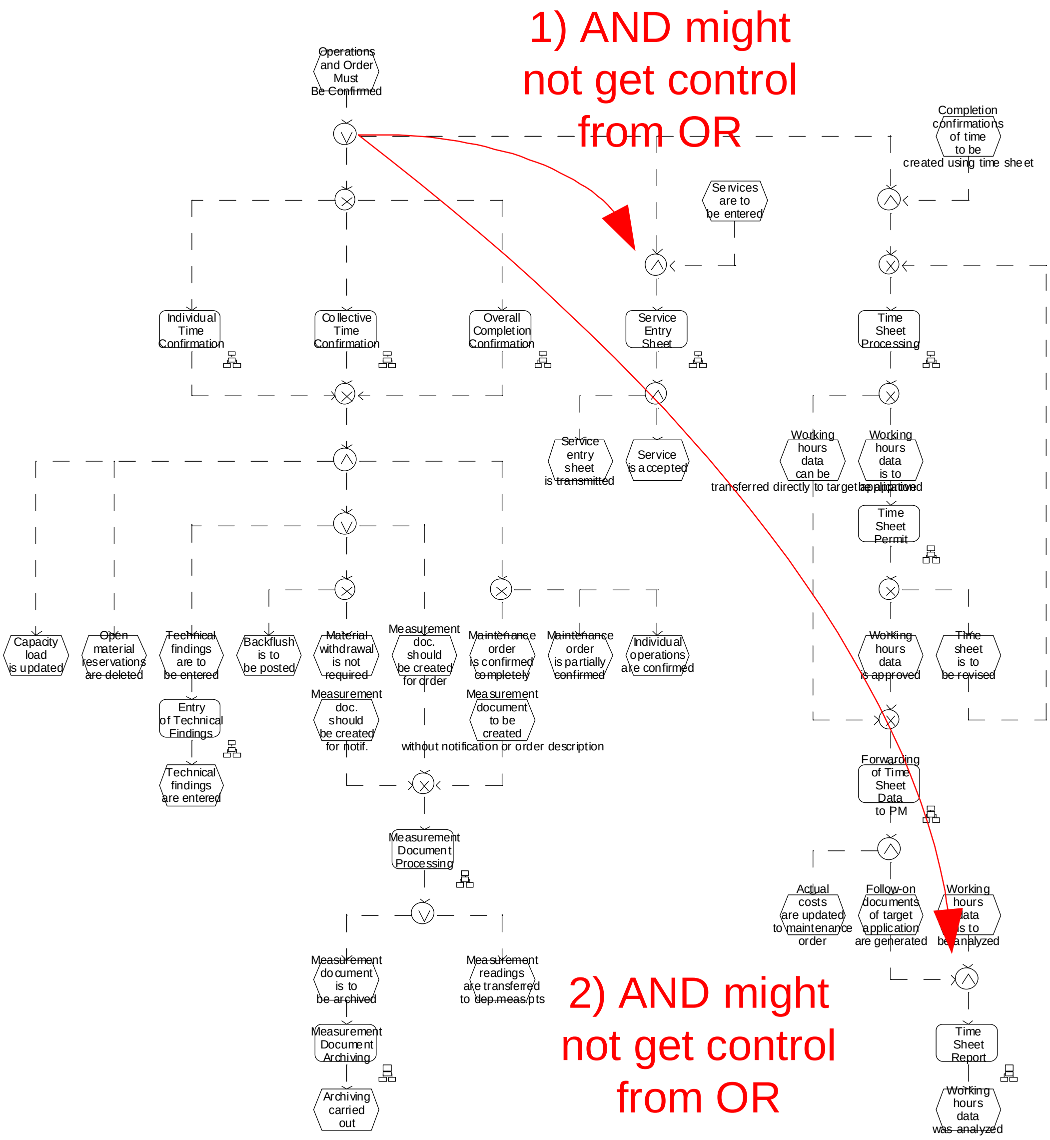


Figure A.41: Plant Maintenance – Breakdown Maintenance Processing – Completion Confirmation (reduced size 11)

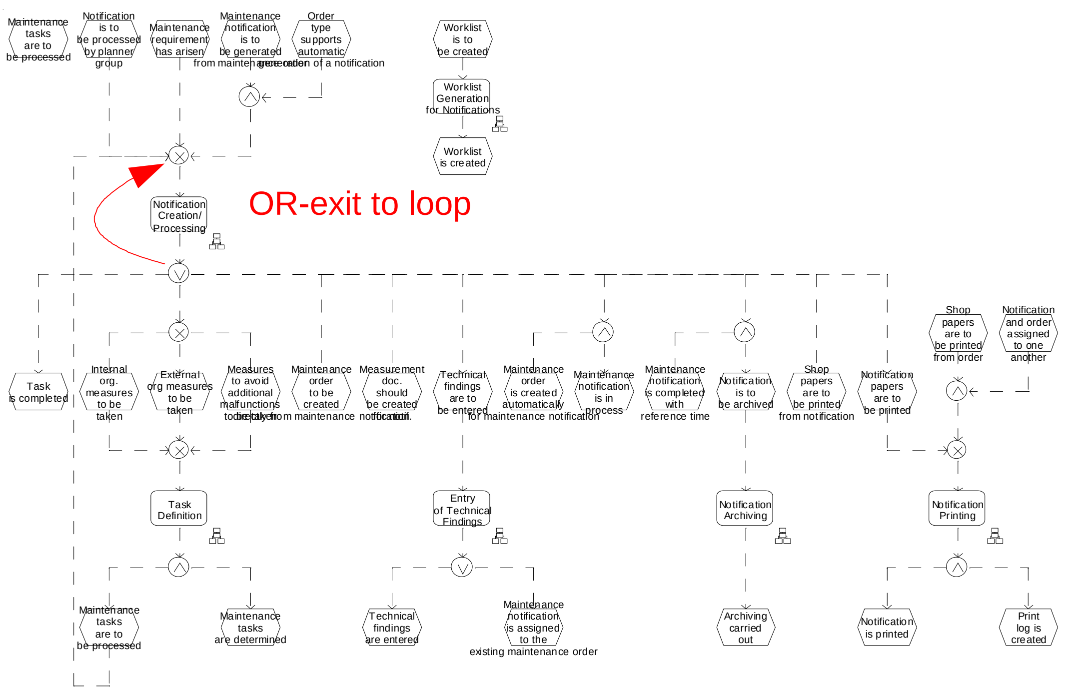


Figure A.42: Plant Maintenance – Breakdown Maintenance Processing – Notification (reduced size 6)

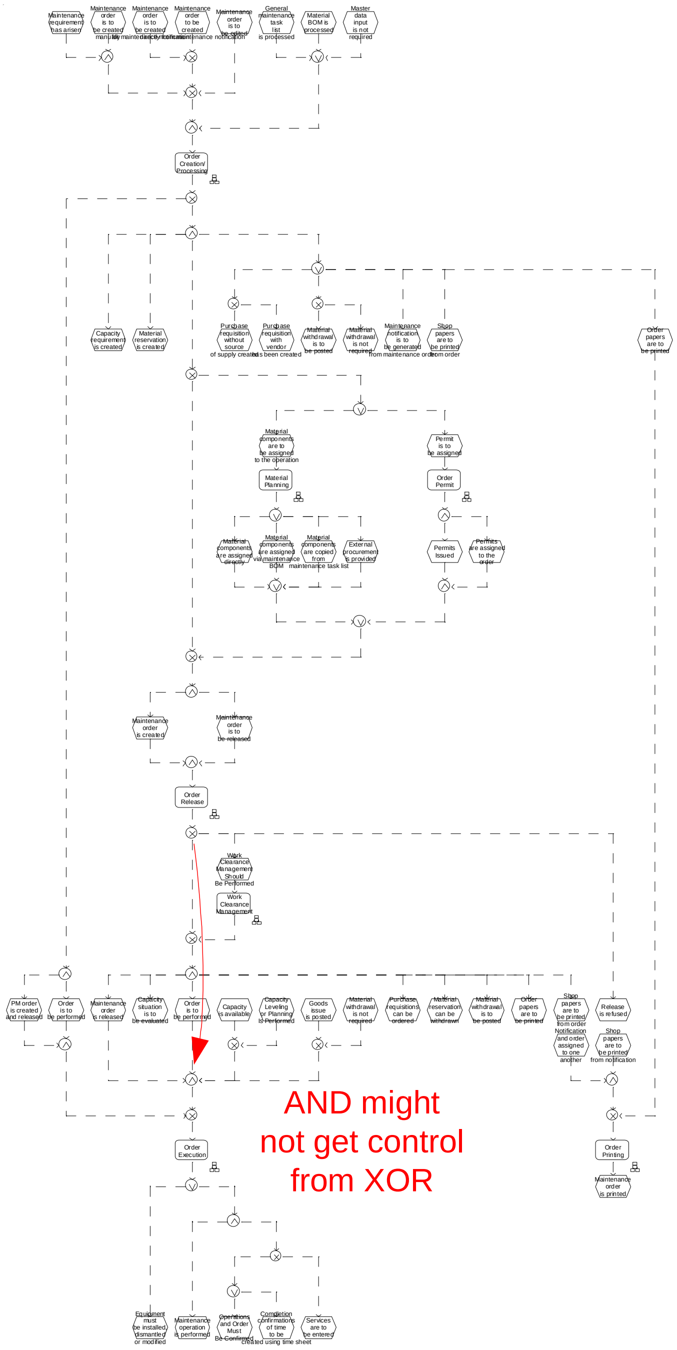


Figure A.43: Plant Maintenance – Breakdown Maintenance Processing – Order (reduced size 12)

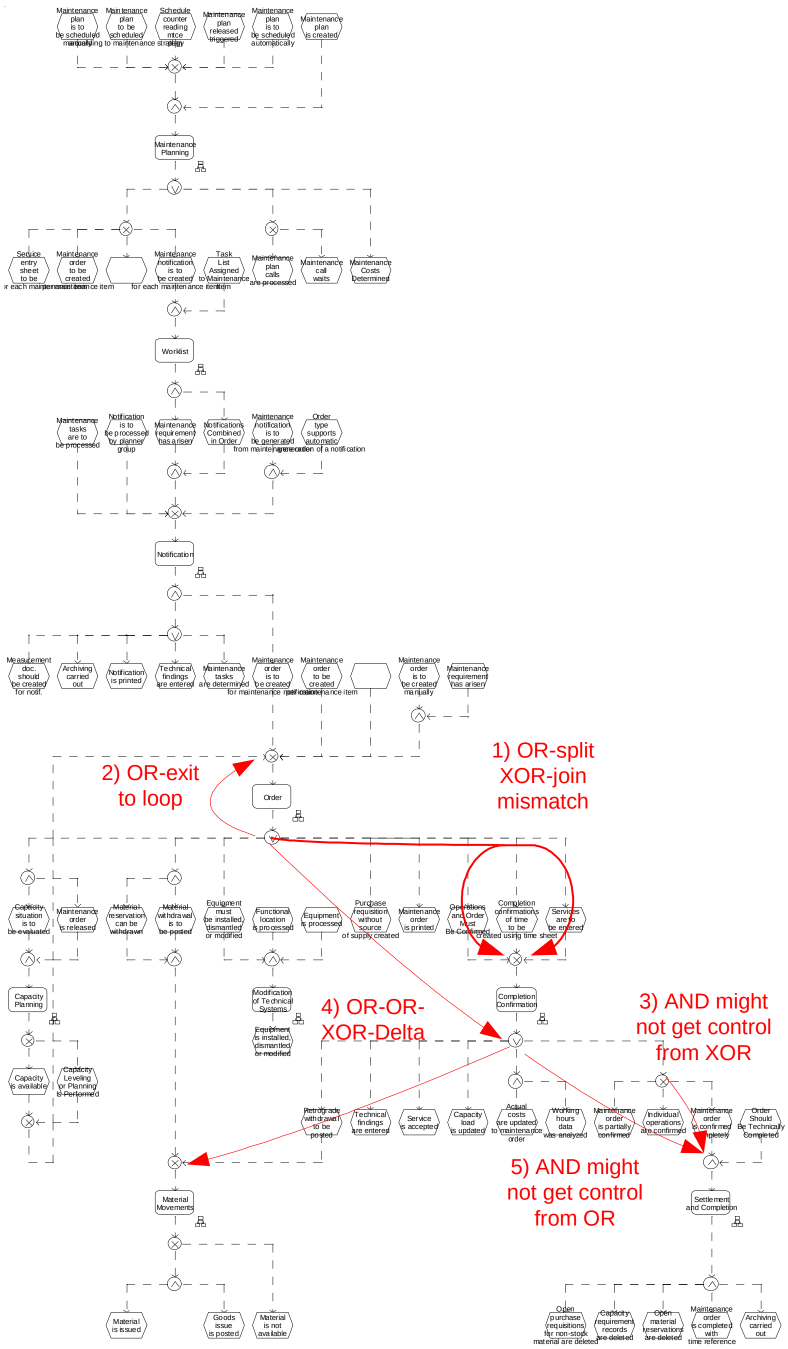


Figure A.44: Plant Maintenance – Planned Maintenance Processing (reduced size 9)

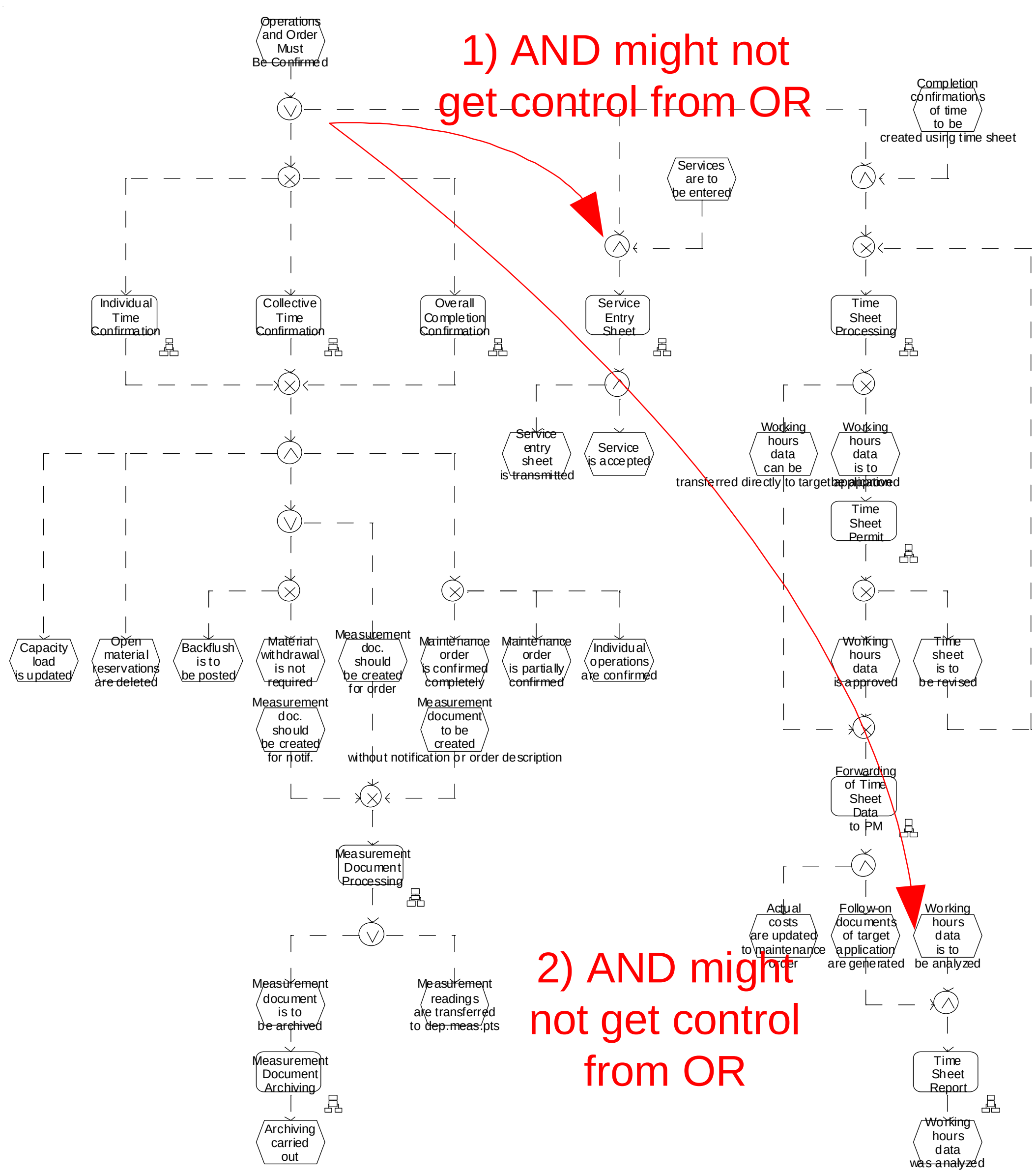


Figure A.45: Plant Maintenance – Planned Maintenance Processing – Completion Confirmation (reduced size 11)

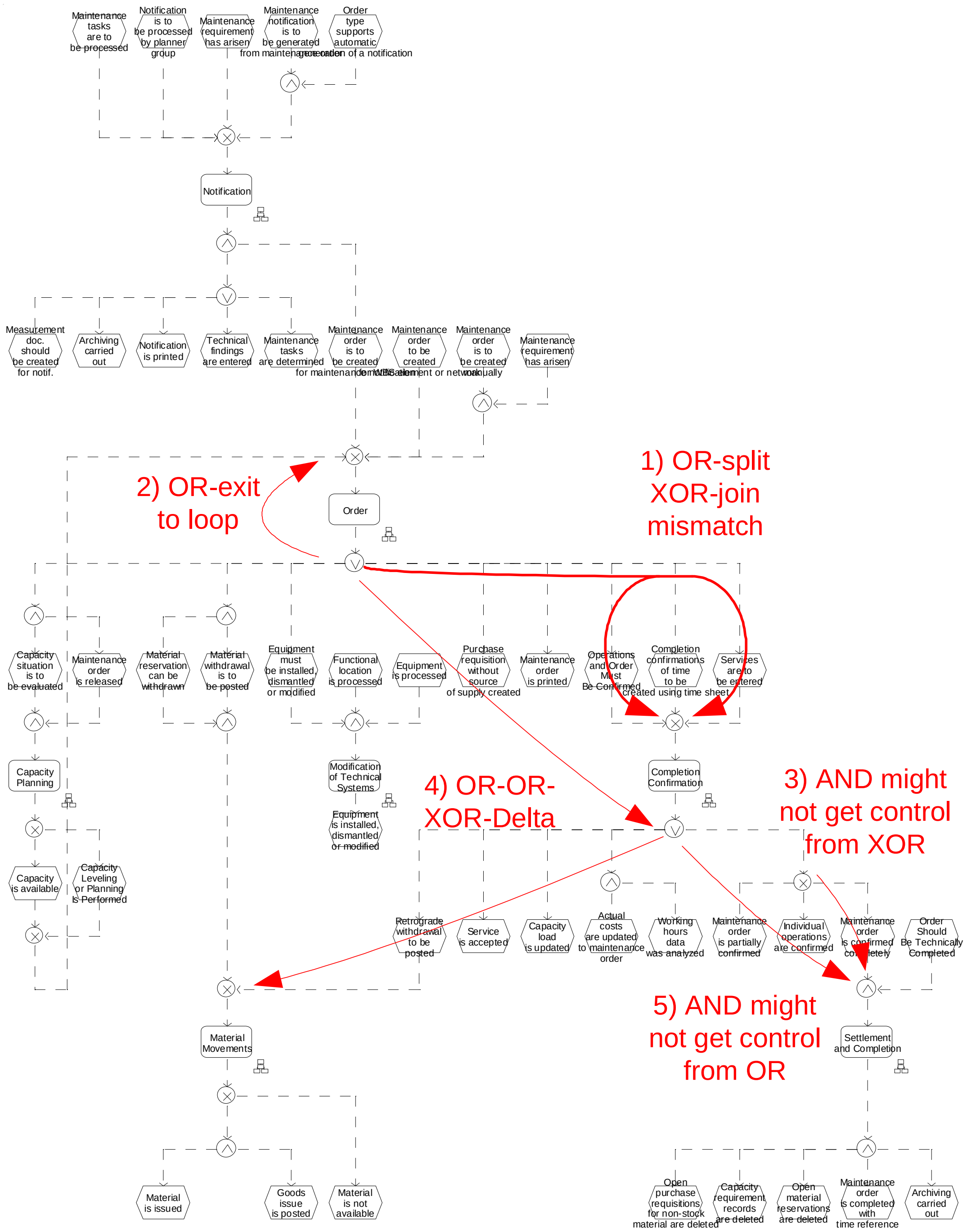


Figure A.46: Plant Maintenance – Project-Based Maintenance Processing (reduced size 9)

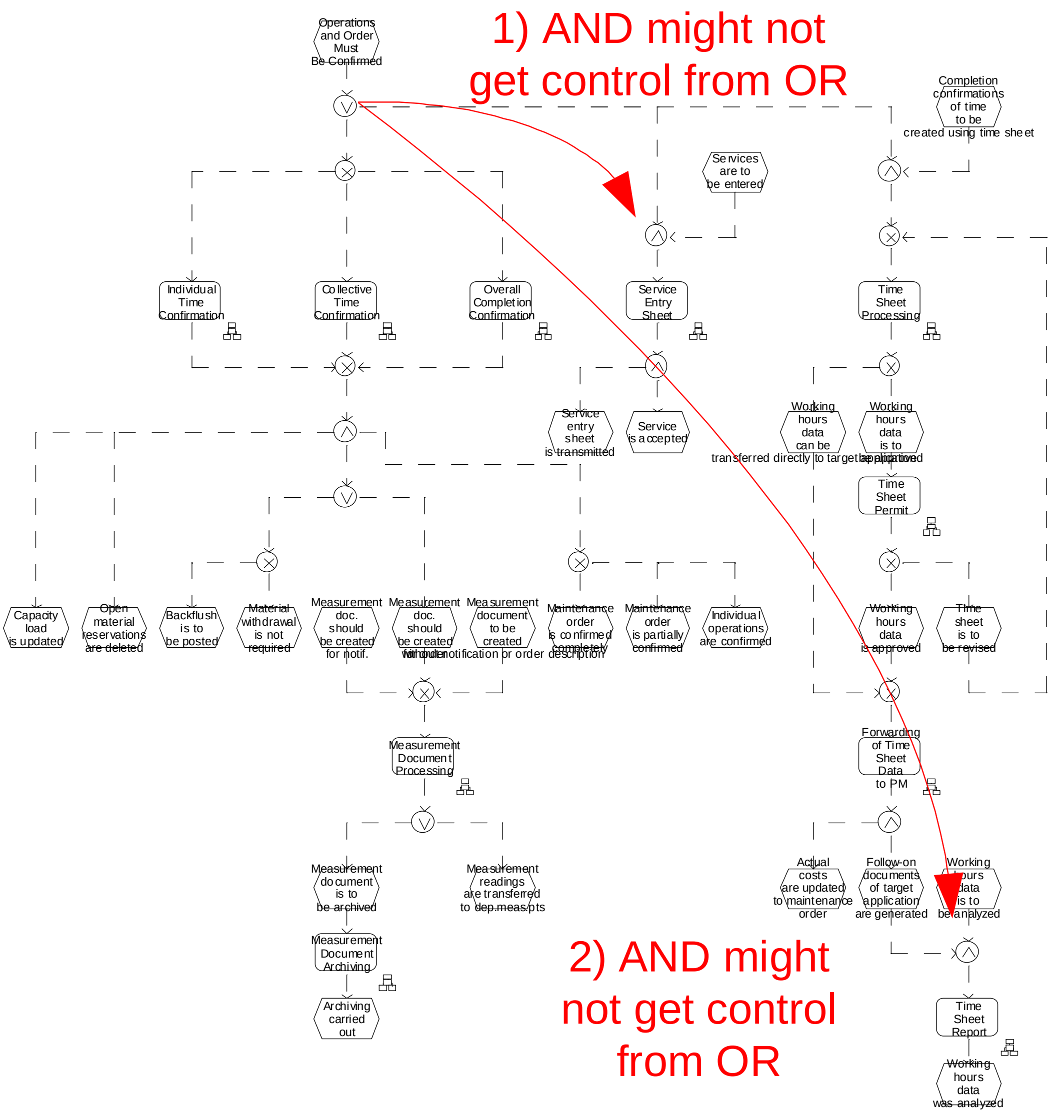


Figure A.47: Plant Maintenance – Project-Based Maintenance Processing – Completion Confirmation (reduced size 11)

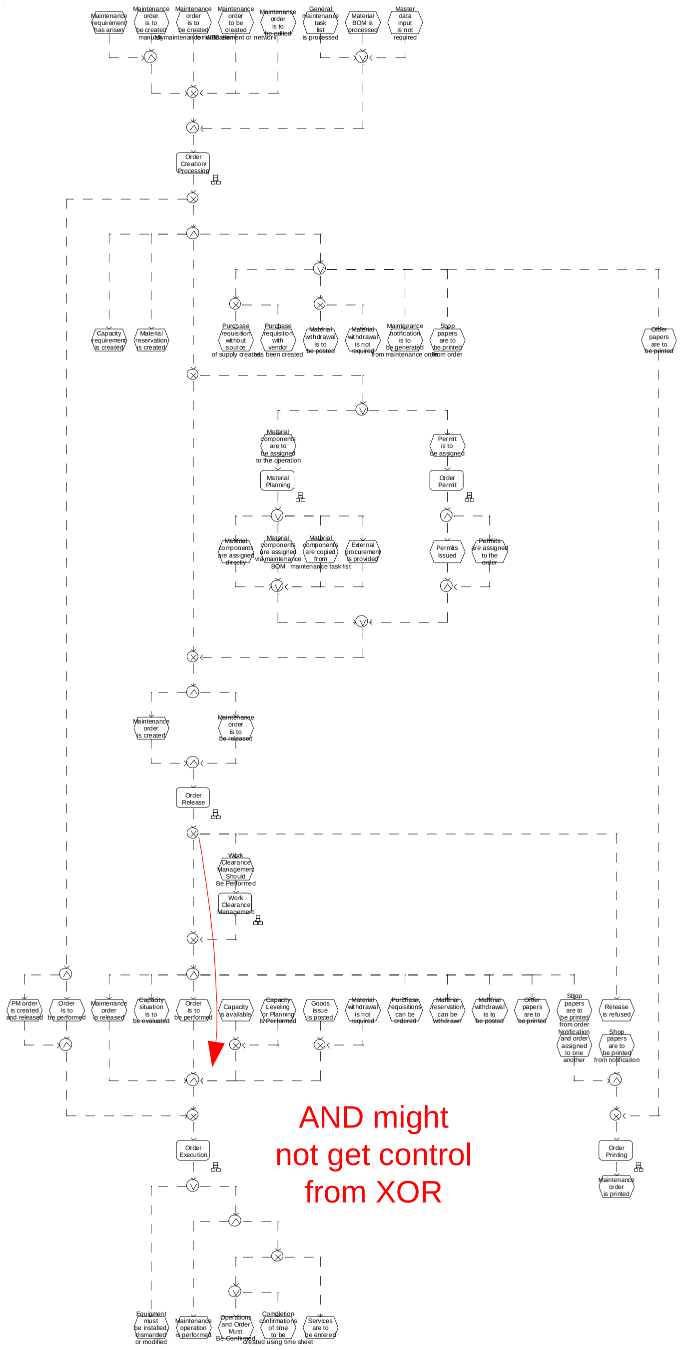


Figure A.48: Plant Maintenance – Project-Based Maintenance Processing – Order (reduced size 12)

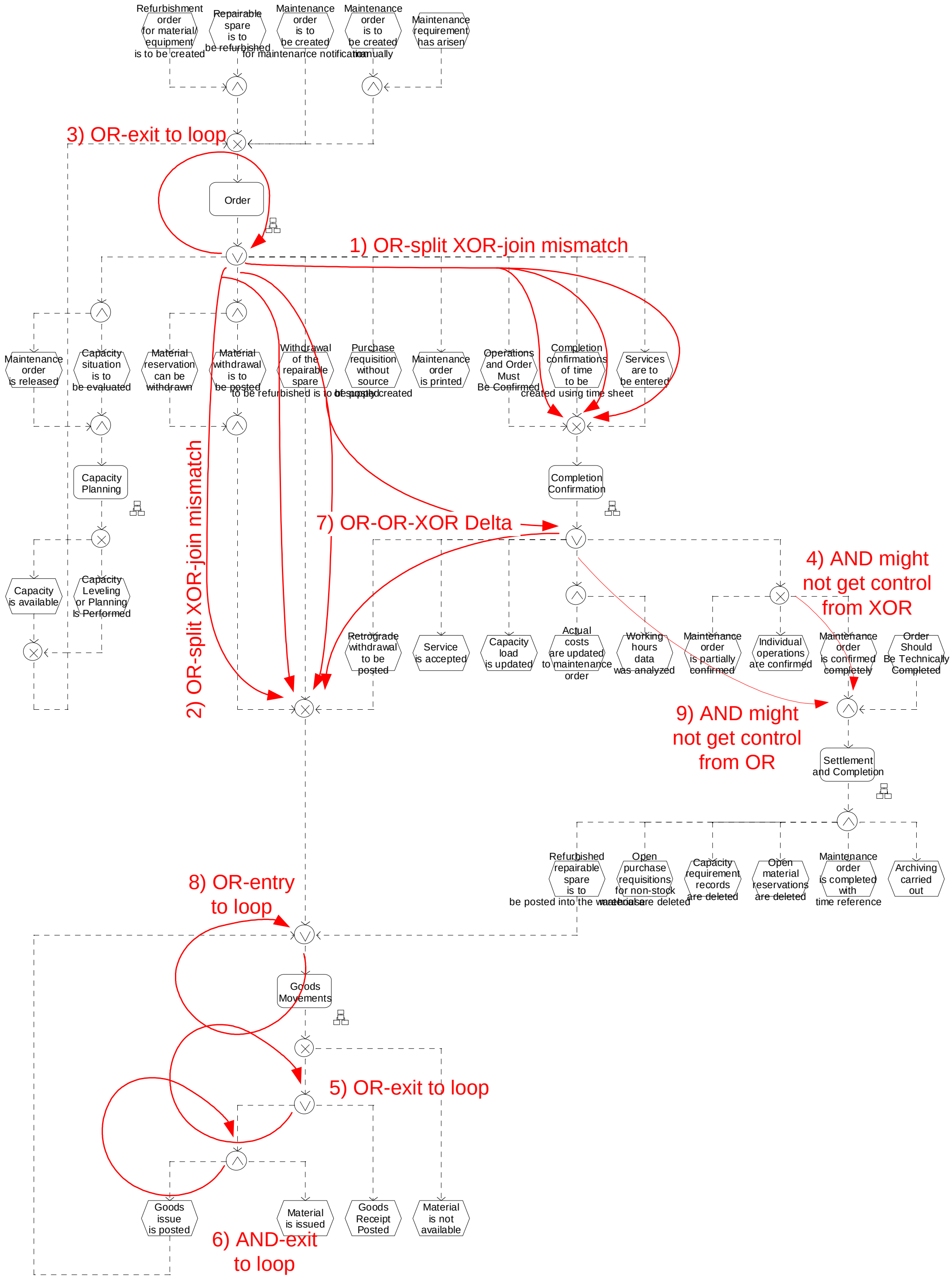


Figure A.49: Plant Maintenance – Refurbishment Processing in Plant Maintenance (reduced size 7)

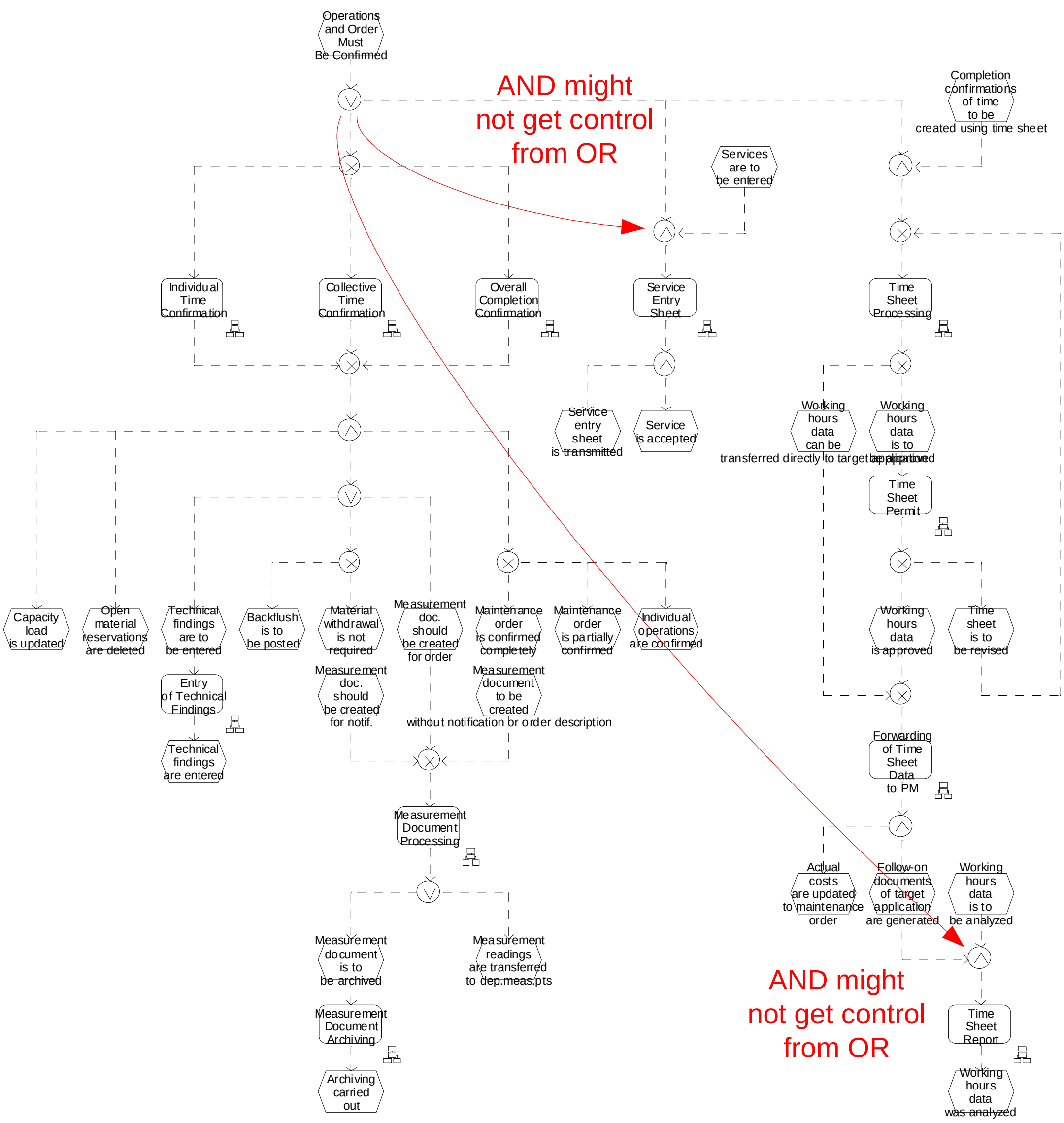


Figure A.50: Plant Maintenance – Refurbishment Processing in Plant Maintenance – Completion Confirmation (reduced size 11)

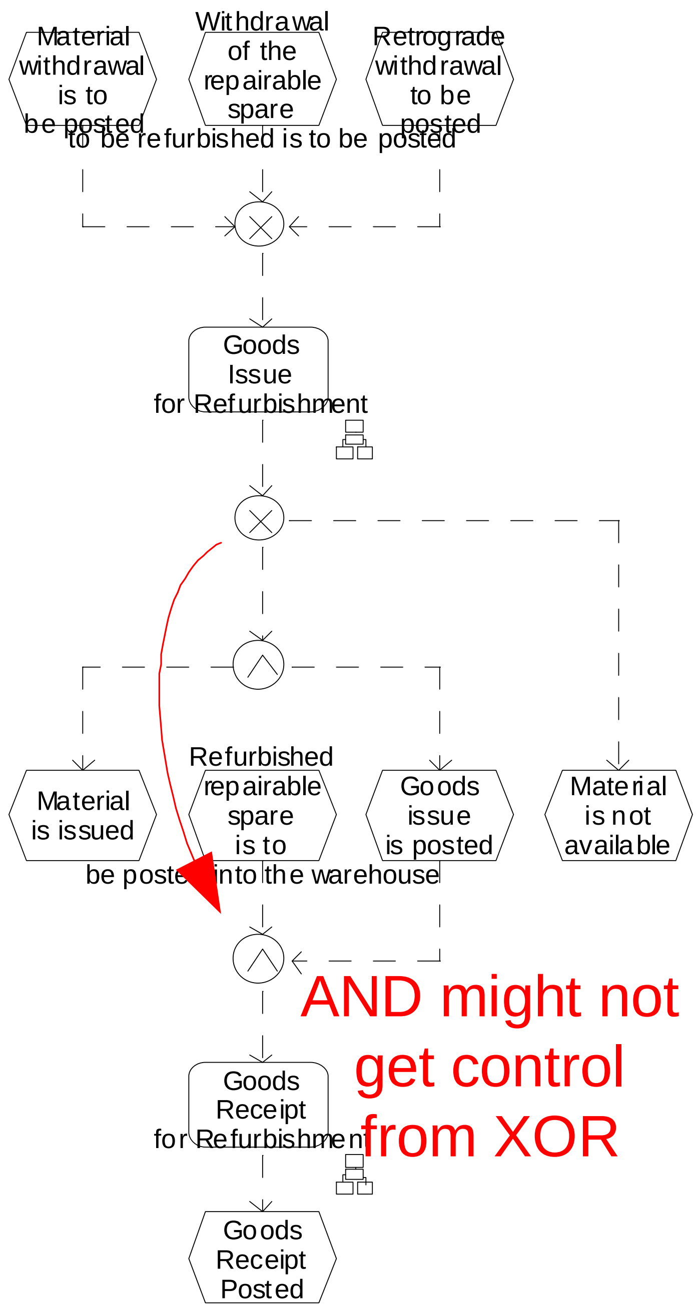


Figure A.51: Plant Maintenance – Refurbishment Processing in Plant Maintenance – Goods Movements (completely reduced)

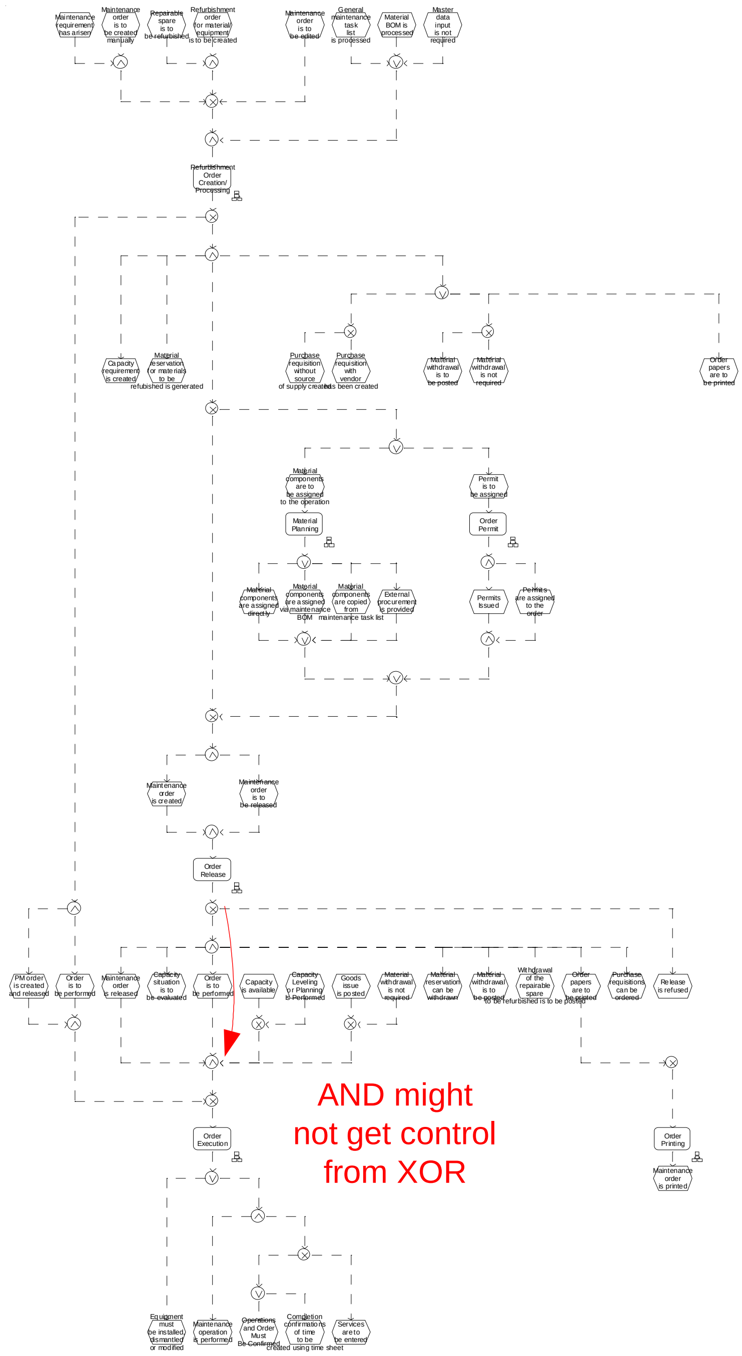


Figure A.52: Plant Maintenance – Refurbishment Processing in Plant Maintenance – Order (reduced size 6)

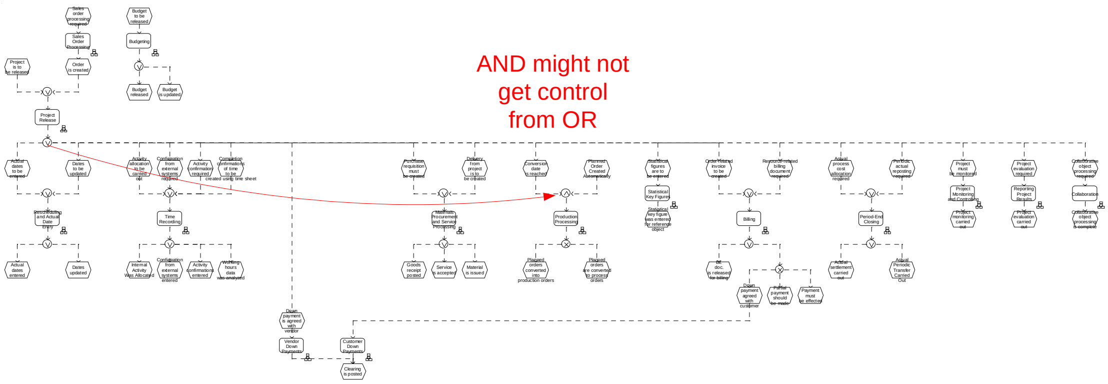


Figure A.53: Project Management – Execution (completely reduced)

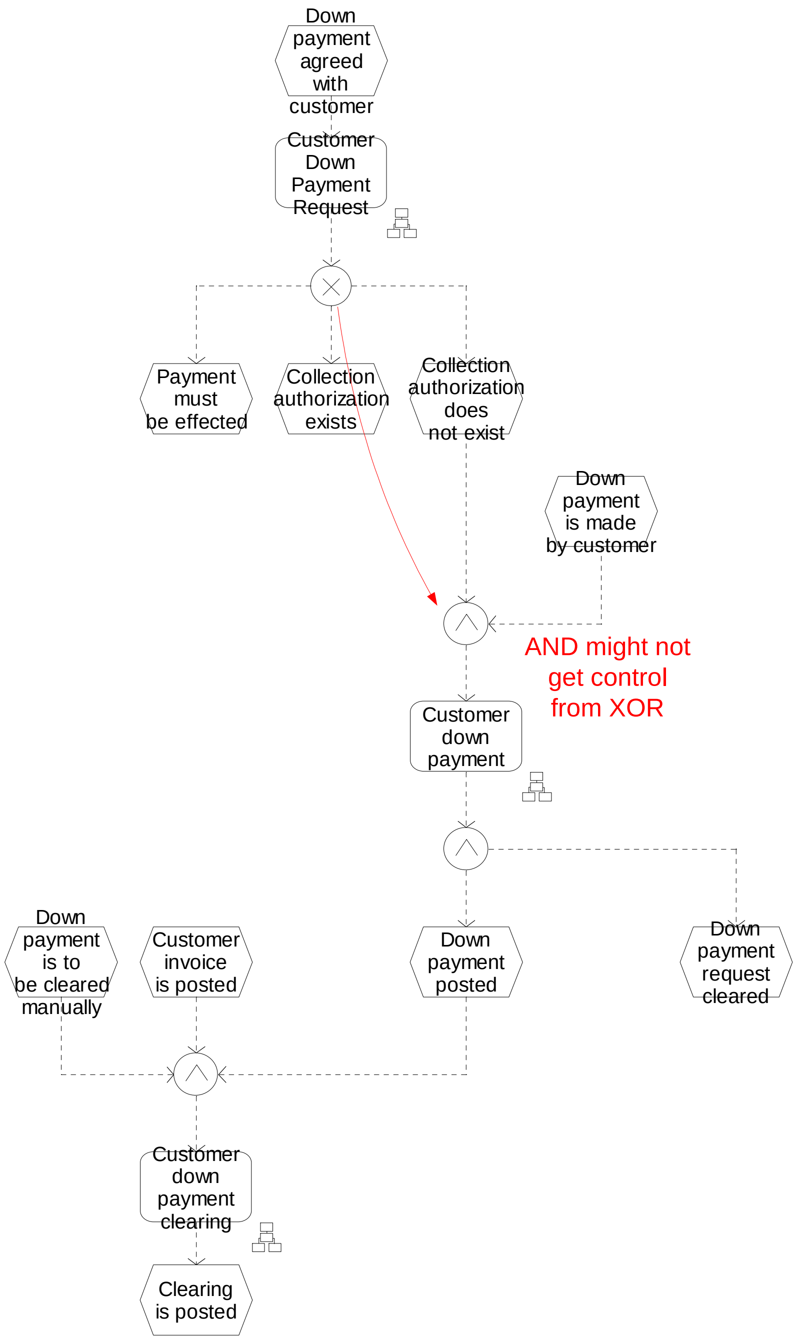


Figure A.54: Project Management – Execution – Customer Down Payments (completely reduced)

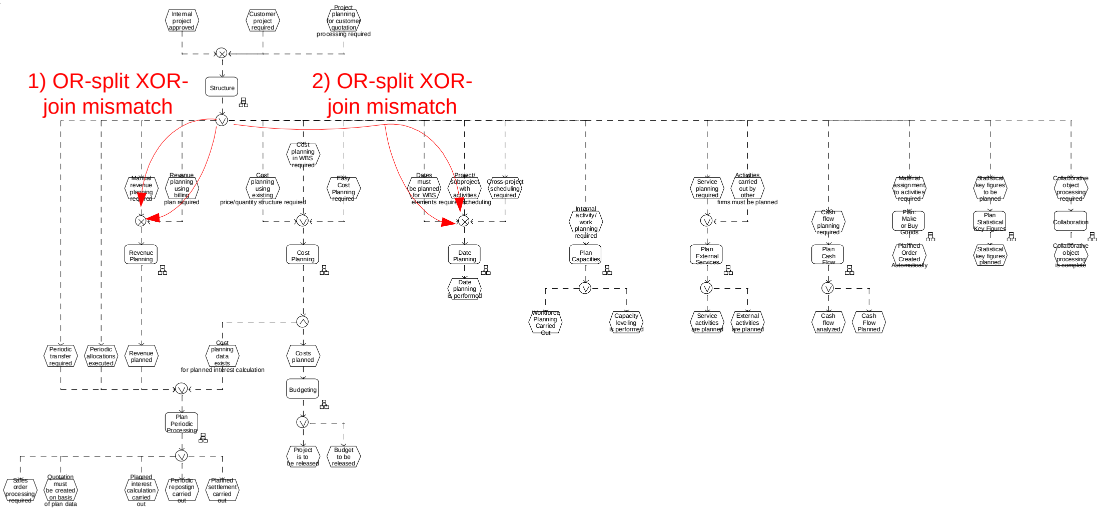


Figure A.55: Project Management – Planning (completely reduced)

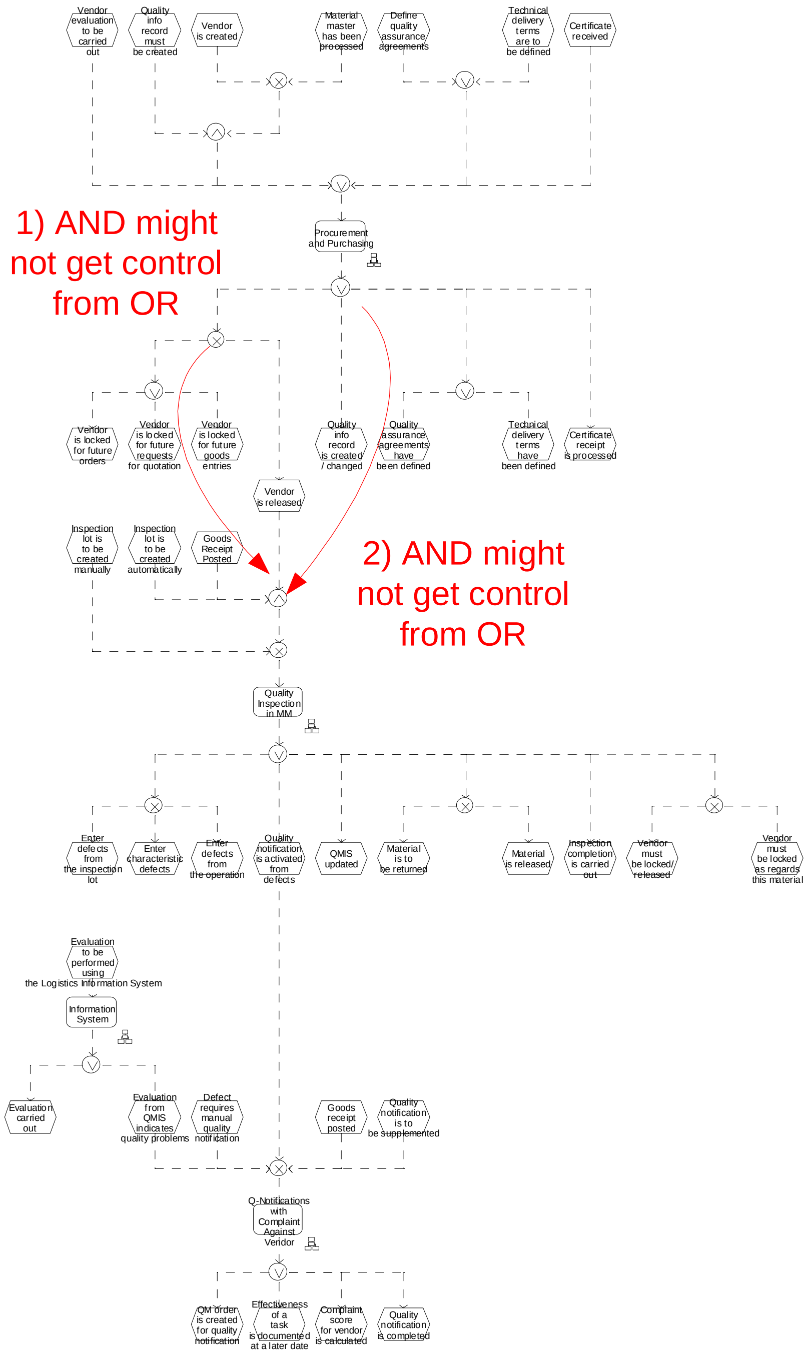


Figure A.56: Quality Management – QM in Materials Management (reduced size 9)

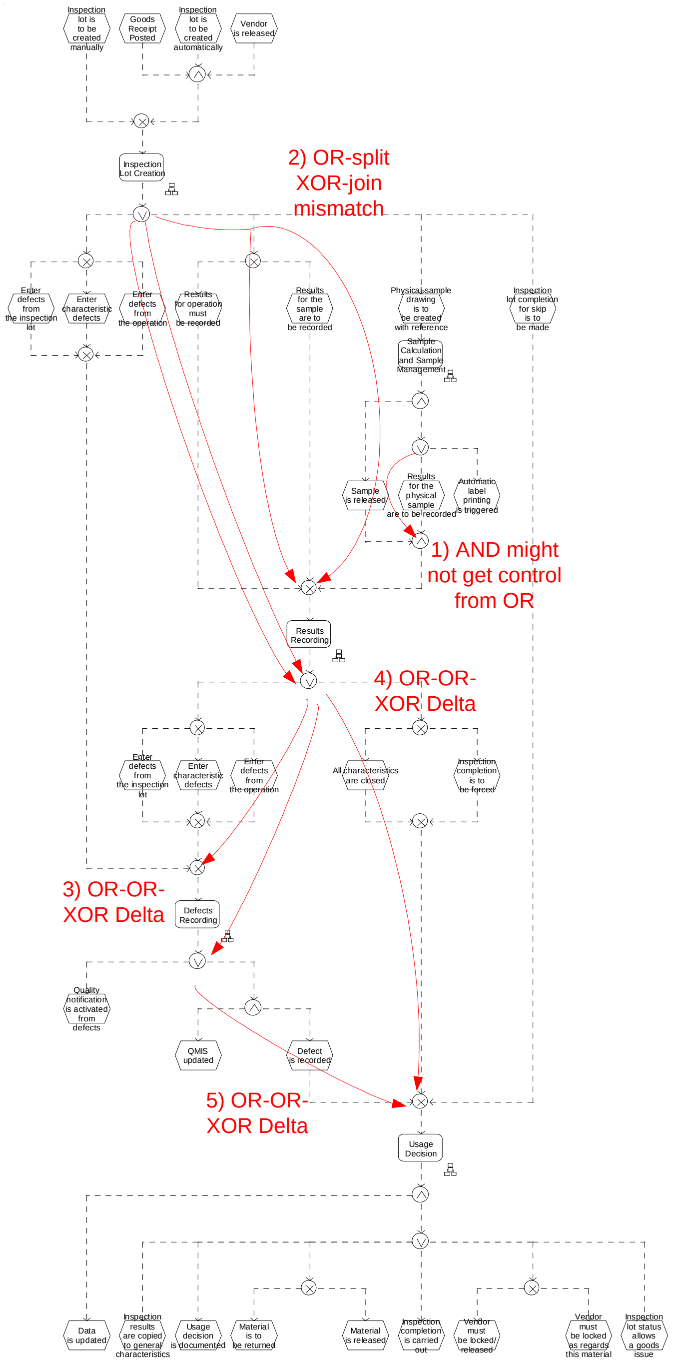


Figure A.57: Quality Management – QM in Materials Management – Quality Inspection in MM (completely reduced)

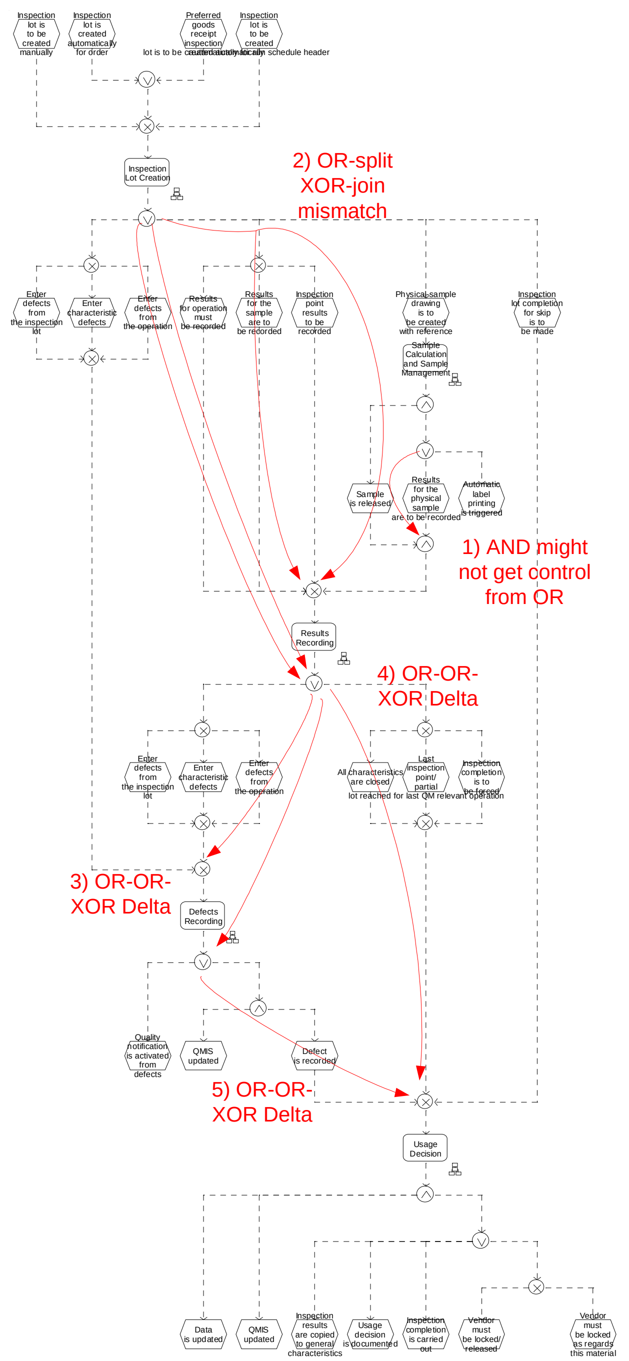


Figure A.58: Quality Management – QM in Production – Inspection During Production (completely reduced)

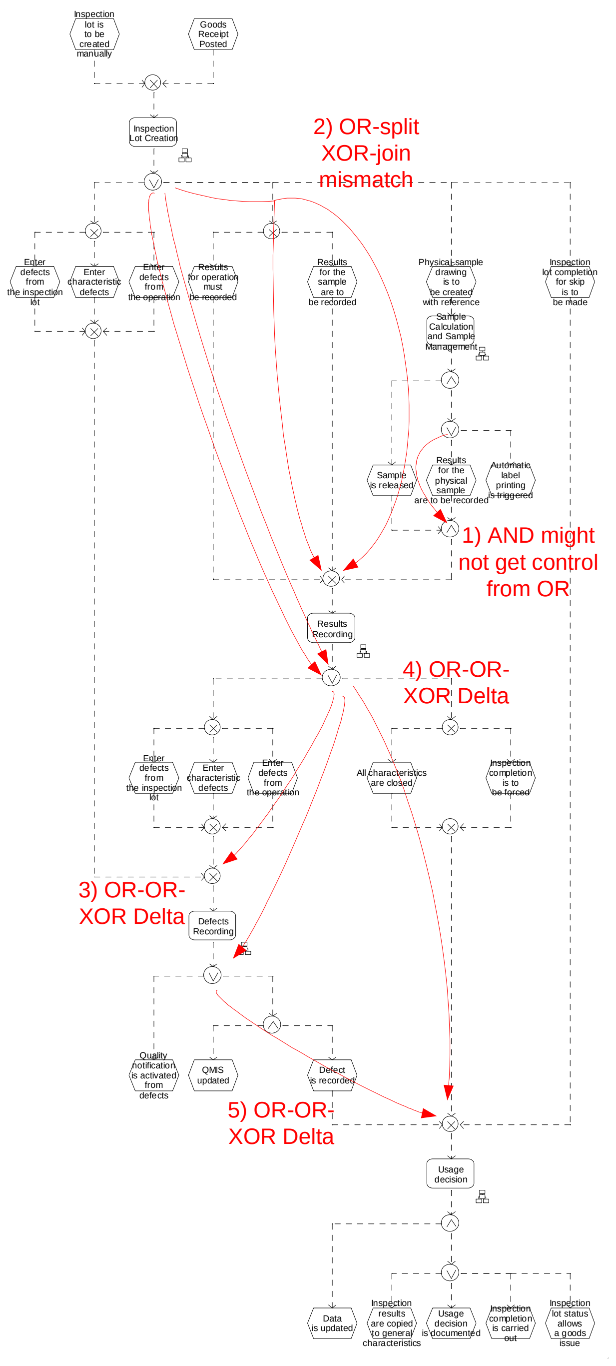


Figure A.59: Quality Management – QM in Production – Quality Inspection for Goods Receipt from Production (completely reduced)

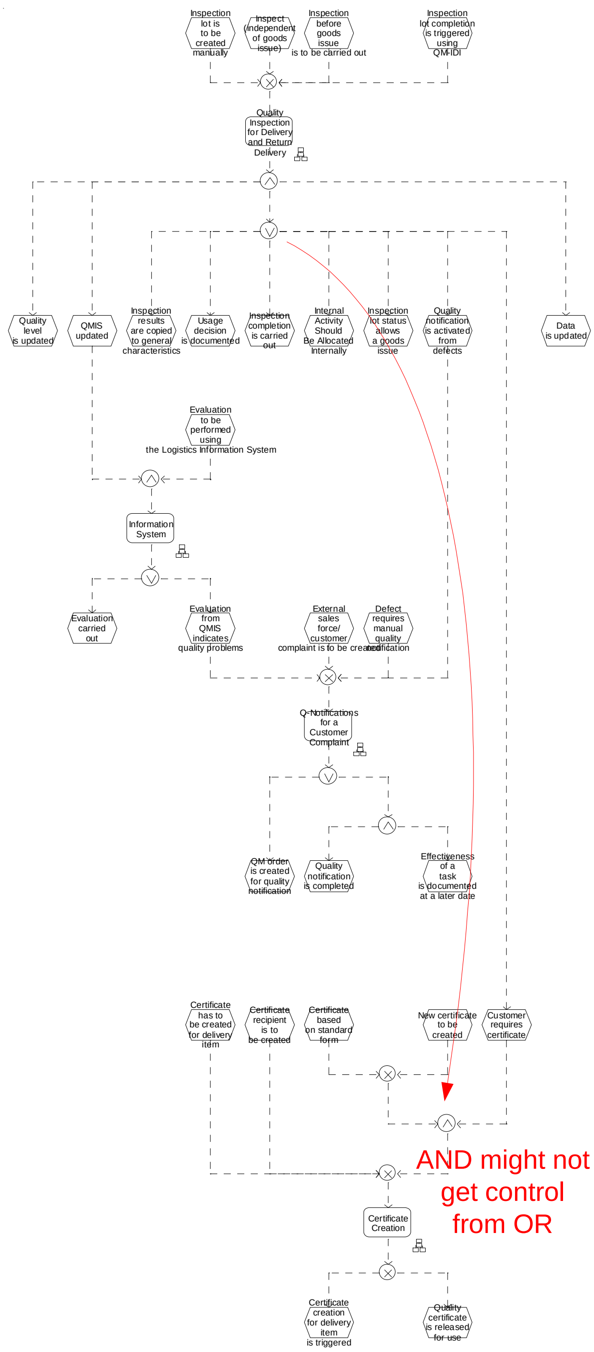


Figure A.60: Quality Management – QM in Sales and Distribution (reduced size 15)

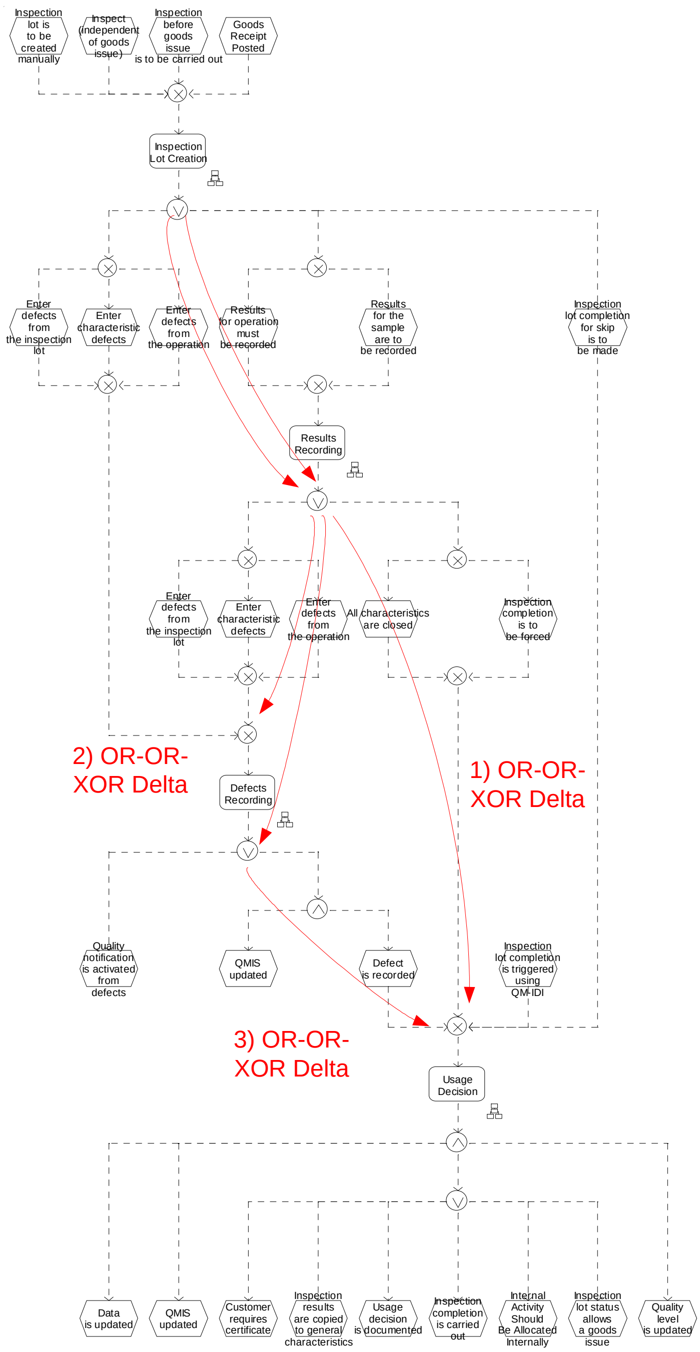


Figure A.61: Quality Management – QM in Sales and Distribution – Quality Inspection for Delivery and Return Delivery (reduced size 6)

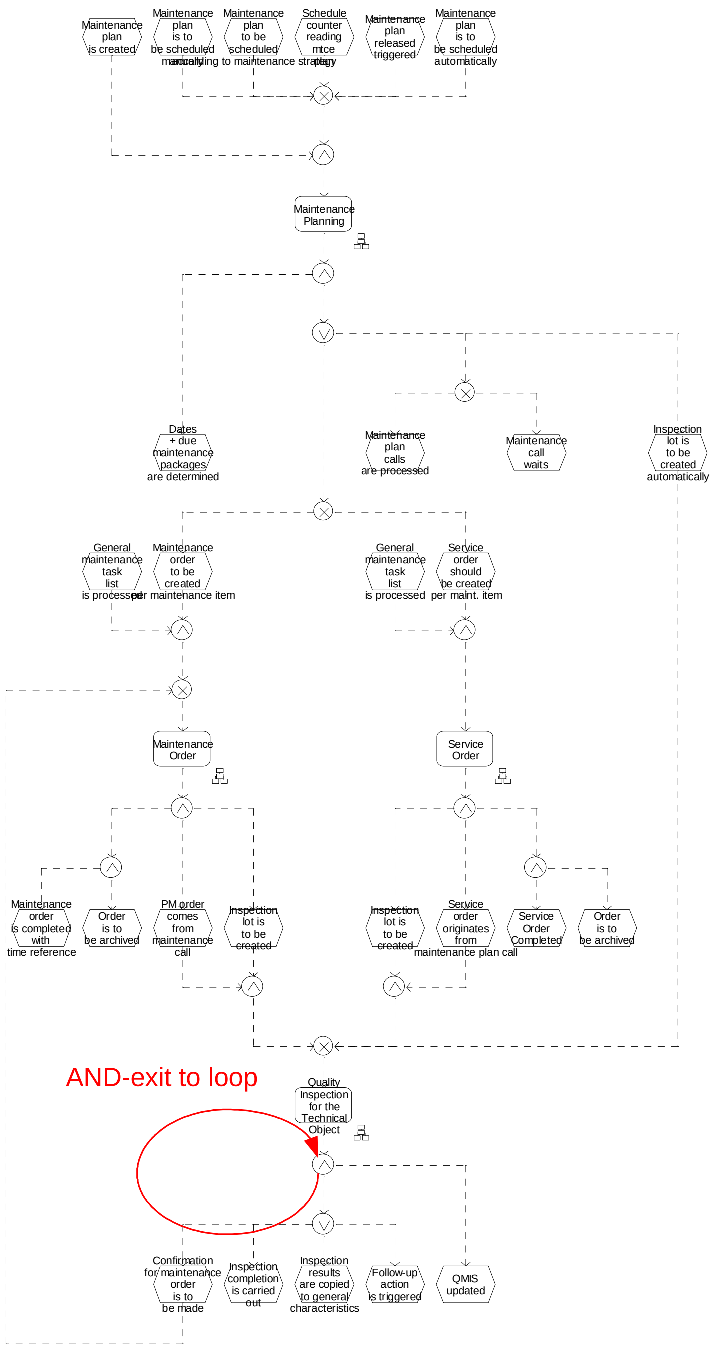


Figure A.62: Quality Management – Test Equipment Management (reduced size 14)

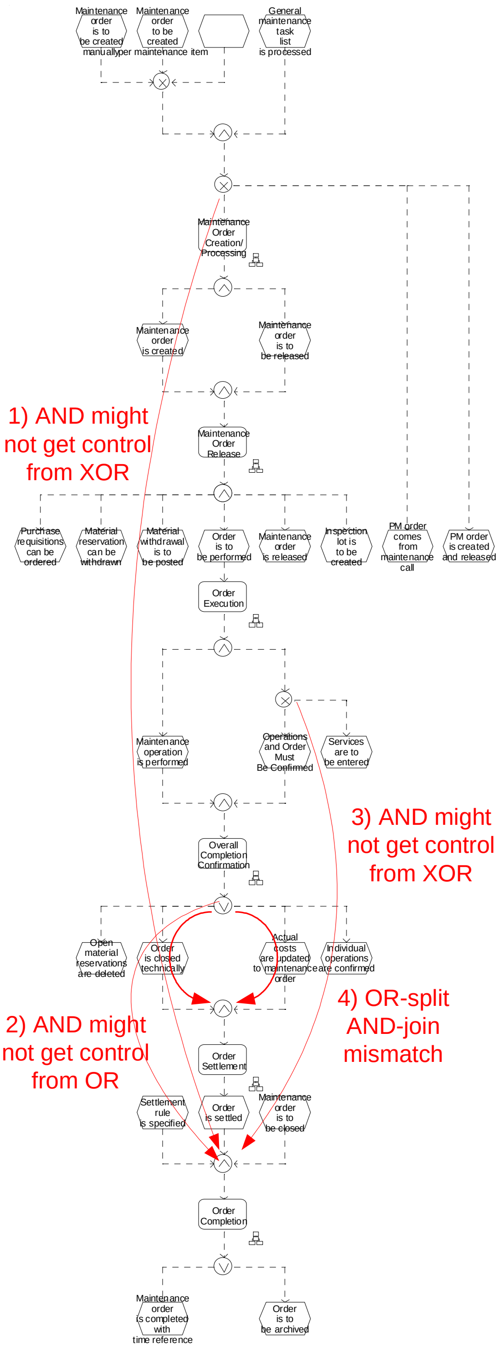


Figure A.63: Quality Management – Test Equipment Management – Maintenance Order (completely reduced)

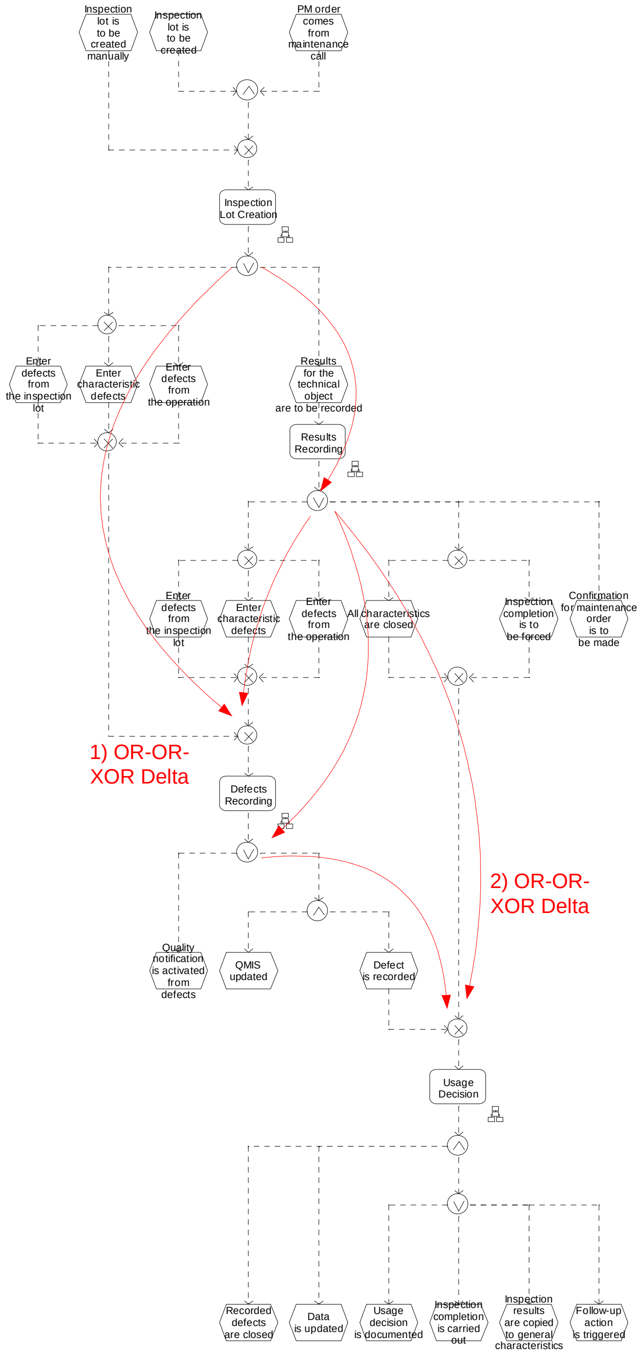


Figure A.64: Quality Management – Test Equipment Management – Quality Inspection for the Technical Object (completely reduced)

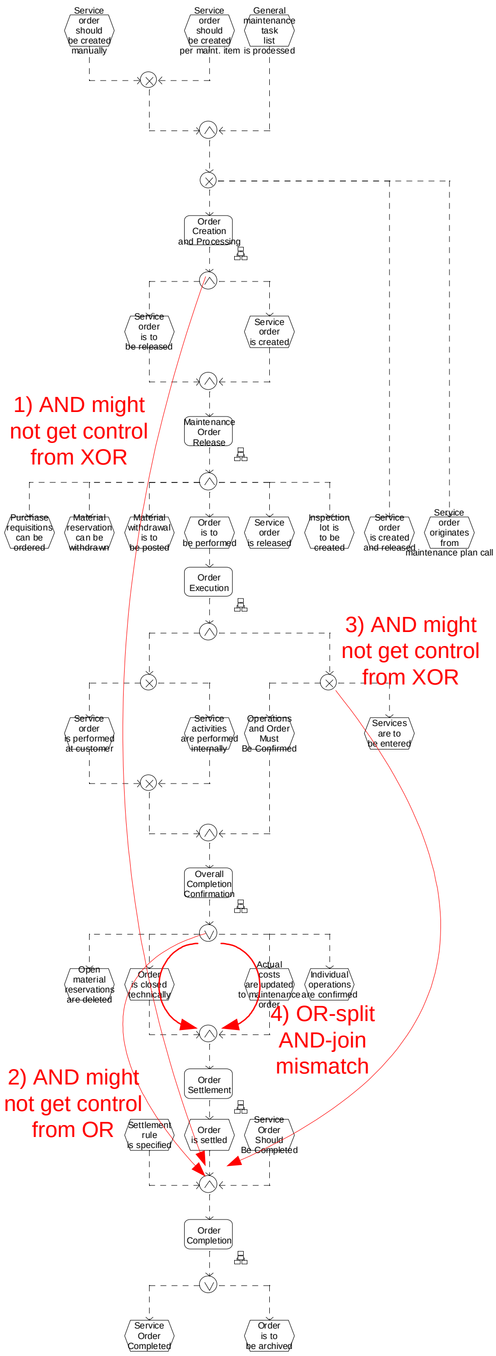


Figure A.65: Quality Management – Test Equipment Management – Service Order (completely reduced)

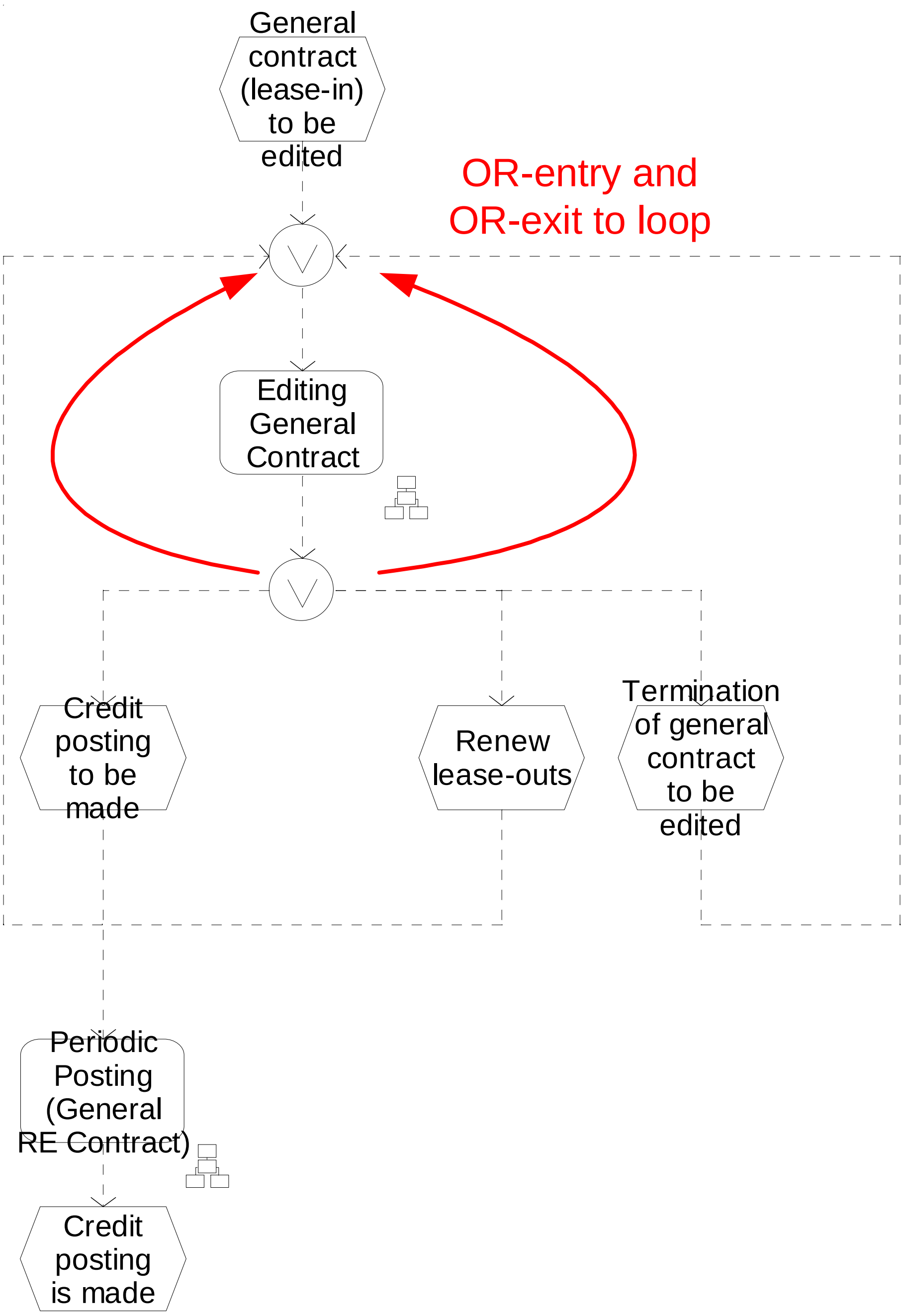


Figure A.66: Real Estate Management – Real Estate Management – General Contract (completely reduced)

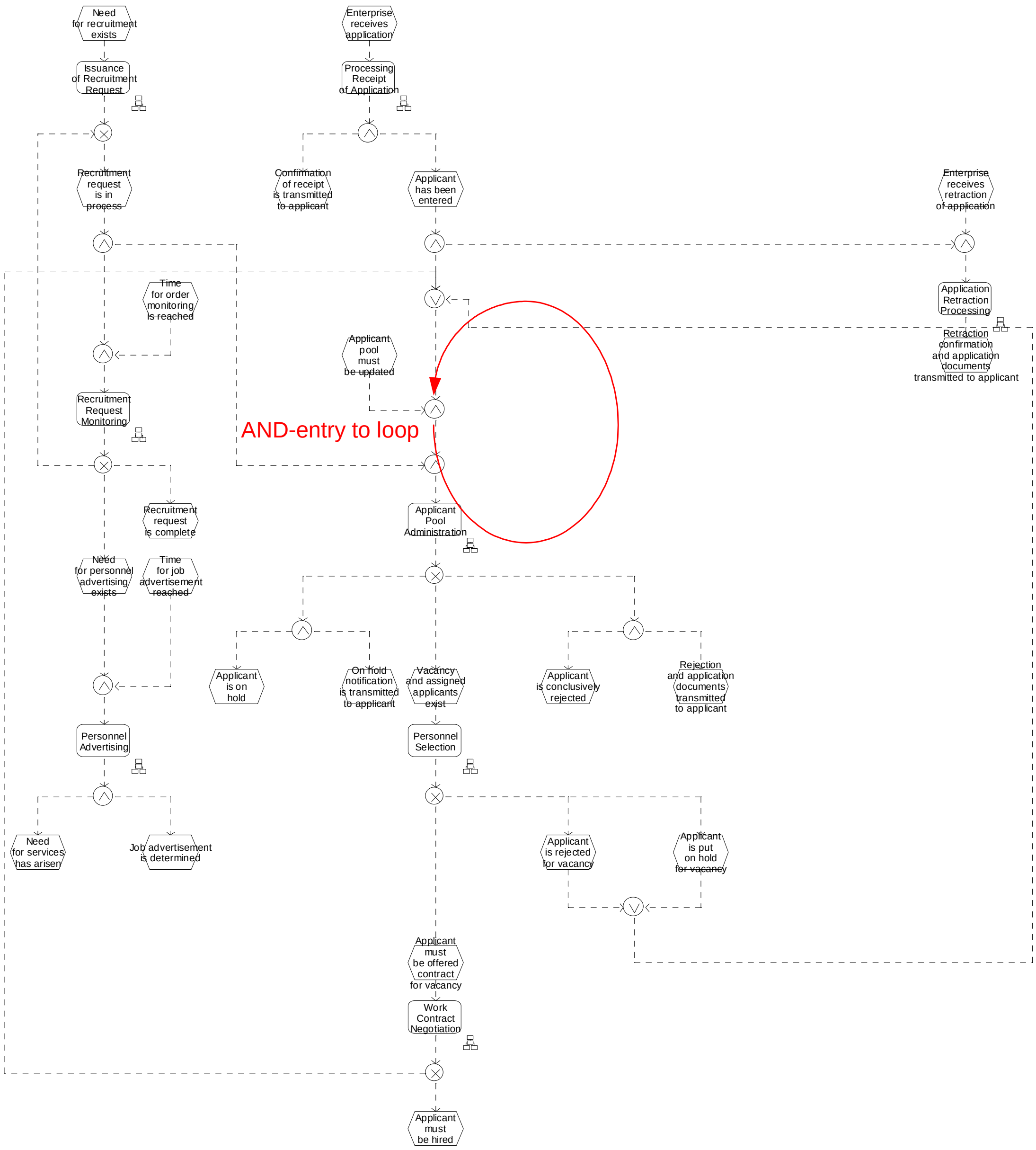


Figure A.67: Recruitment – Recruitment (reduced size 19)

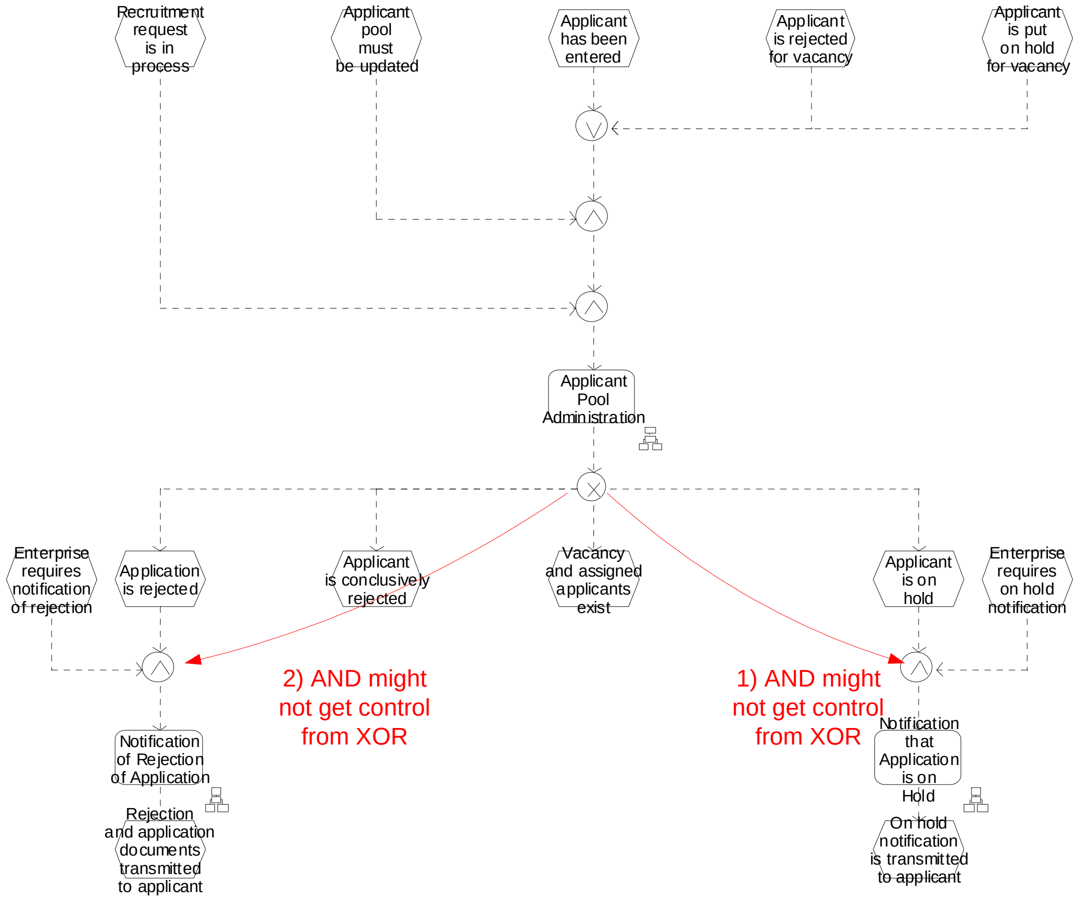


Figure A.68: Recruitment – Recruitment – Applicant Pool Administration (reduced size 8)

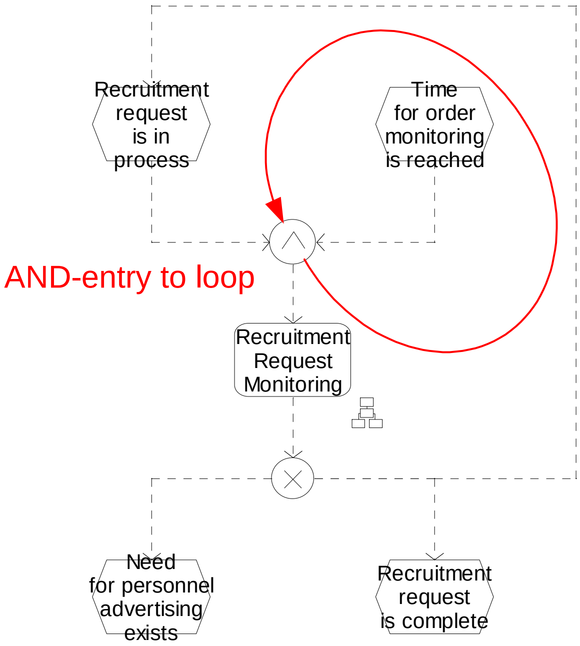


Figure A.69: Recruitment – Recruitment – Recruitment Request Monitoring (completely reduced)

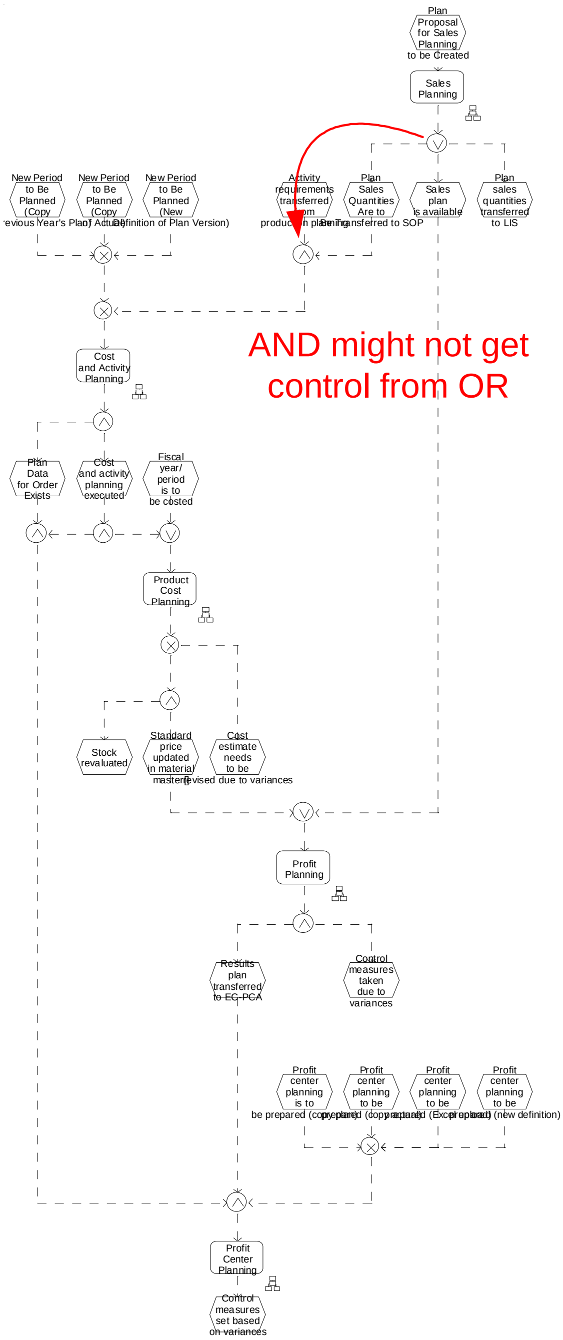


Figure A.70: Revenue and Cost Controlling – Profit and Cost Planning (reduced size 15)

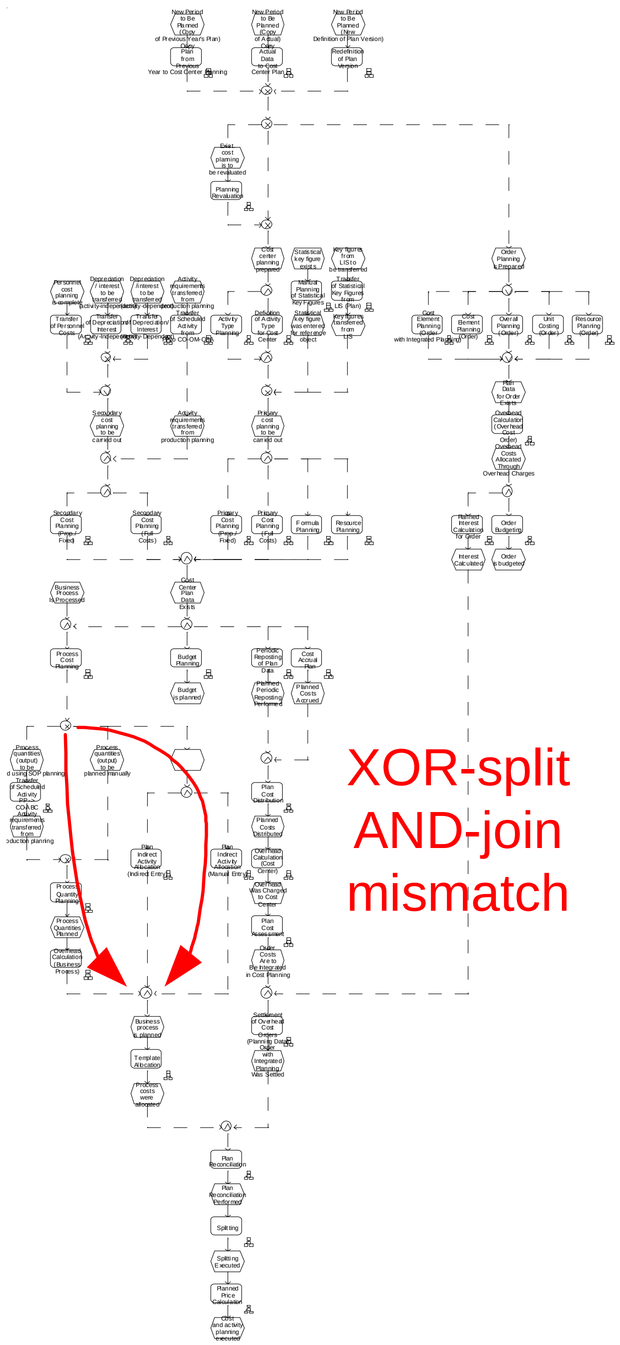


Figure A.71: Revenue and Cost Controlling – Profit and Cost Planning – Cost and Activity Planning (reduced size 7)

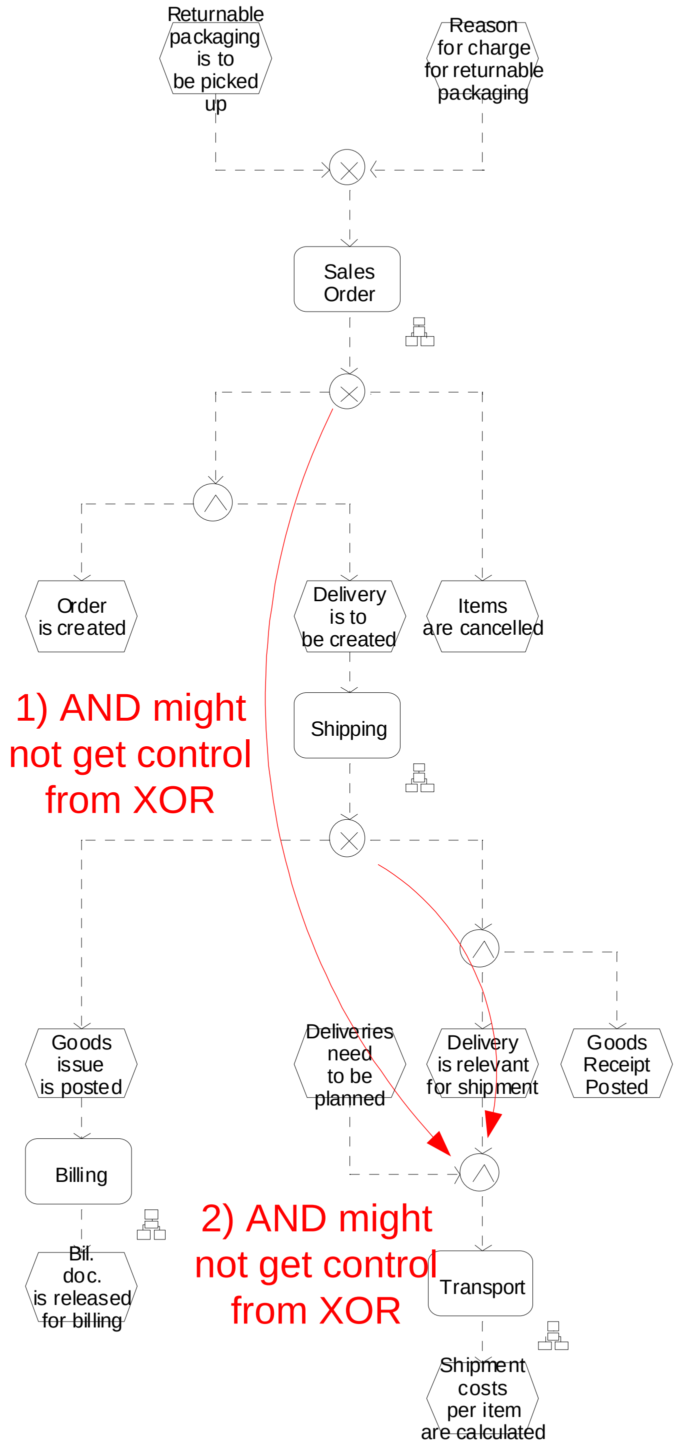


Figure A.72: Sales and Distribution – Empties and Returnable Packaging Handling (completely reduced)

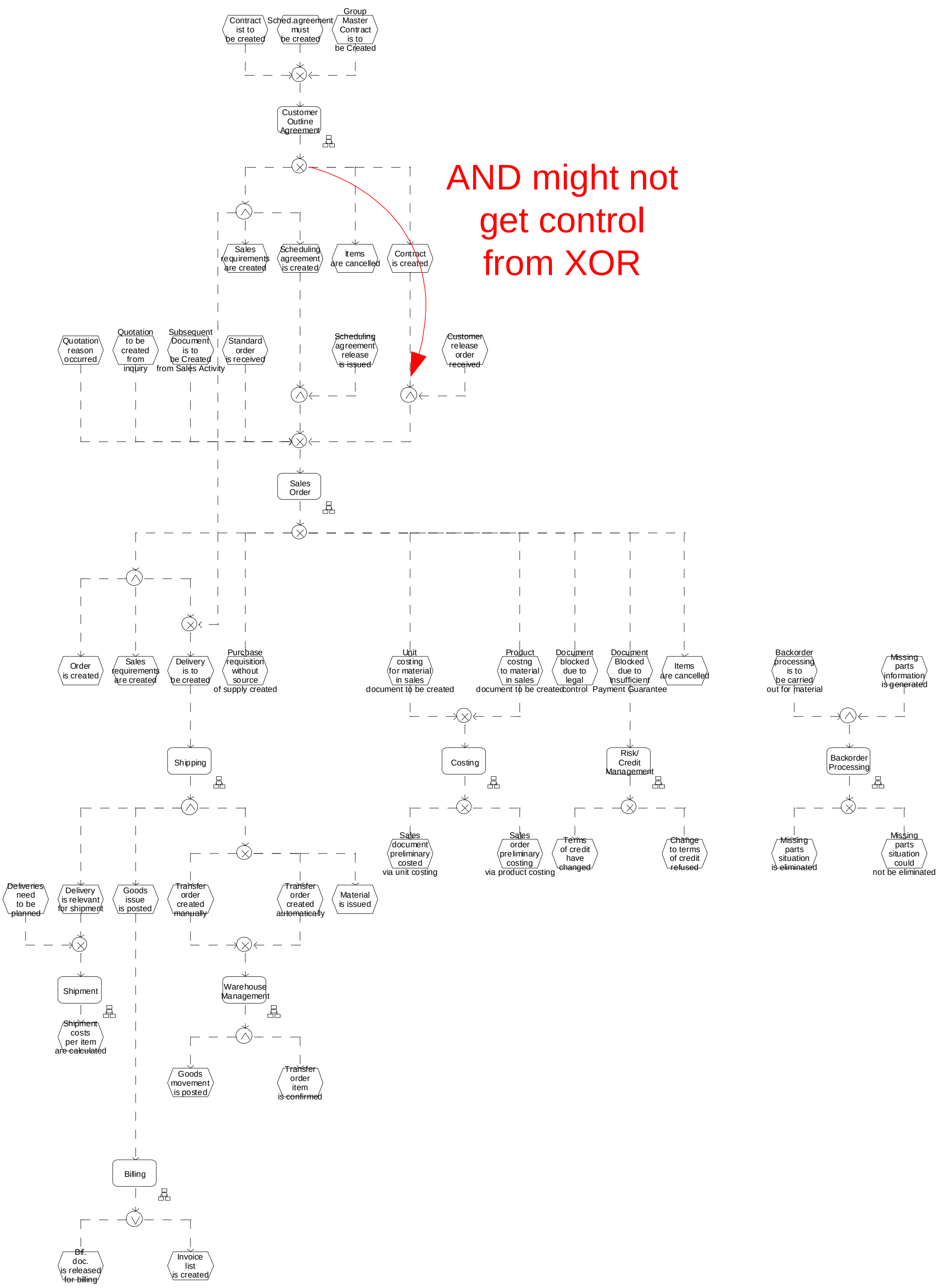


Figure A.73: Sales and Distribution – Sales Order Processing (Standard) (reduced size 14)

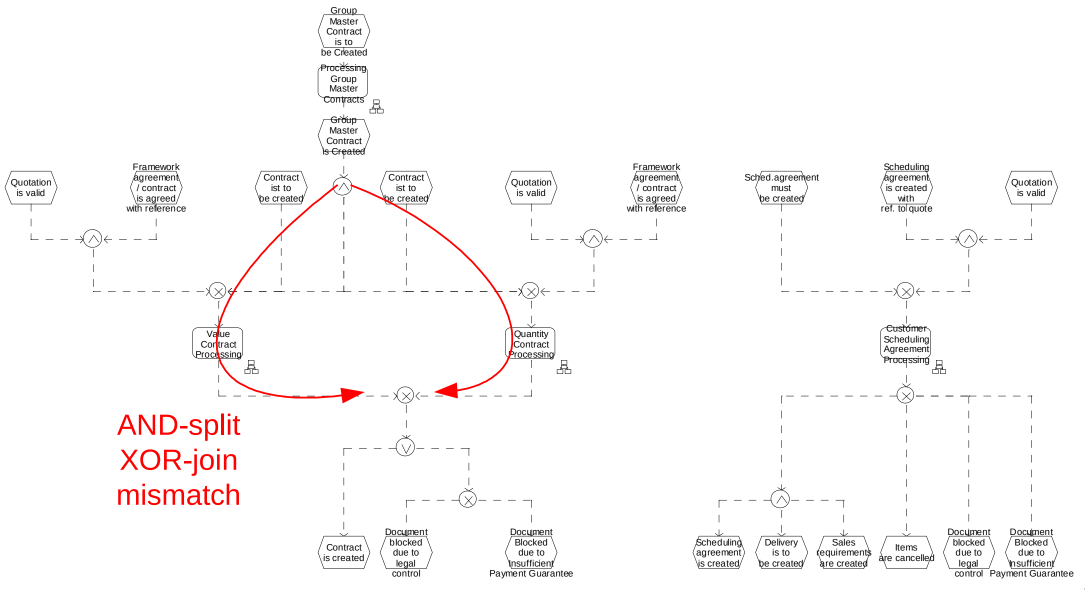


Figure A.74: Sales and Distribution – Sales Order Processing (Standard) – Customer Outline Agreement (completely reduced)

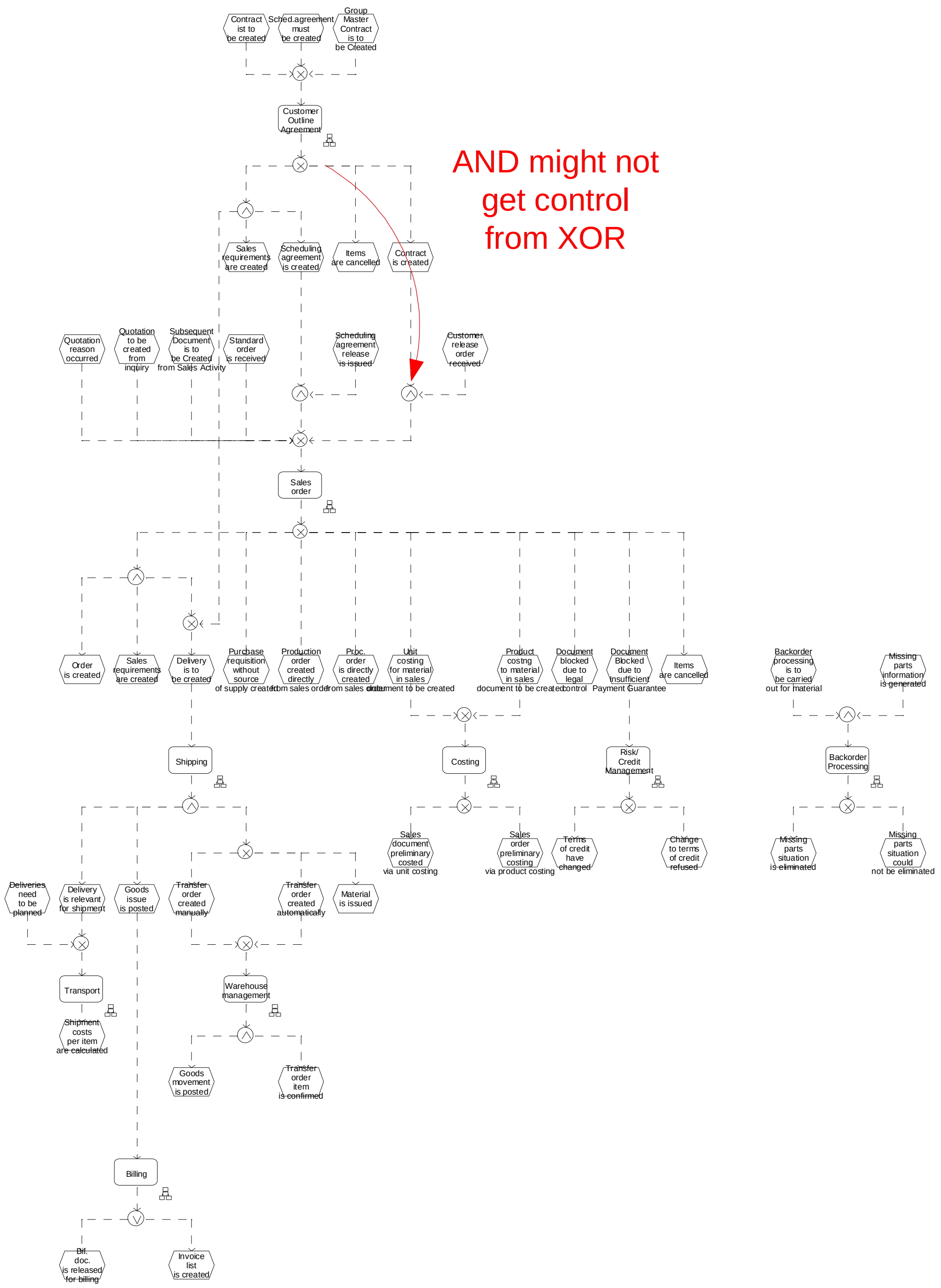


Figure A.75: Sales and Distribution – Sales Order Processing: Make/Assembly To Order (reduced size 14)

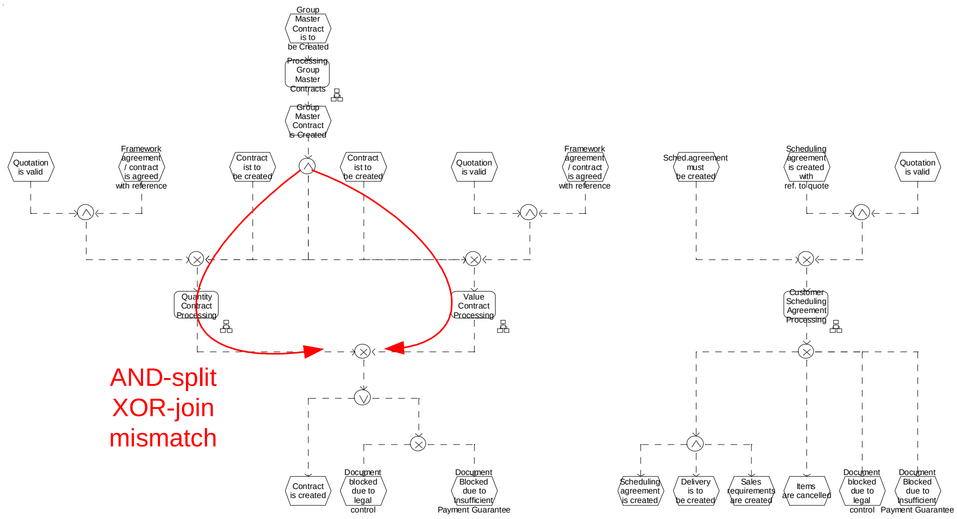


Figure A.76: Sales and Distribution – Sales Order Processing: Make/Assembly To Order – Customer Outline Agreement (completely reduced)

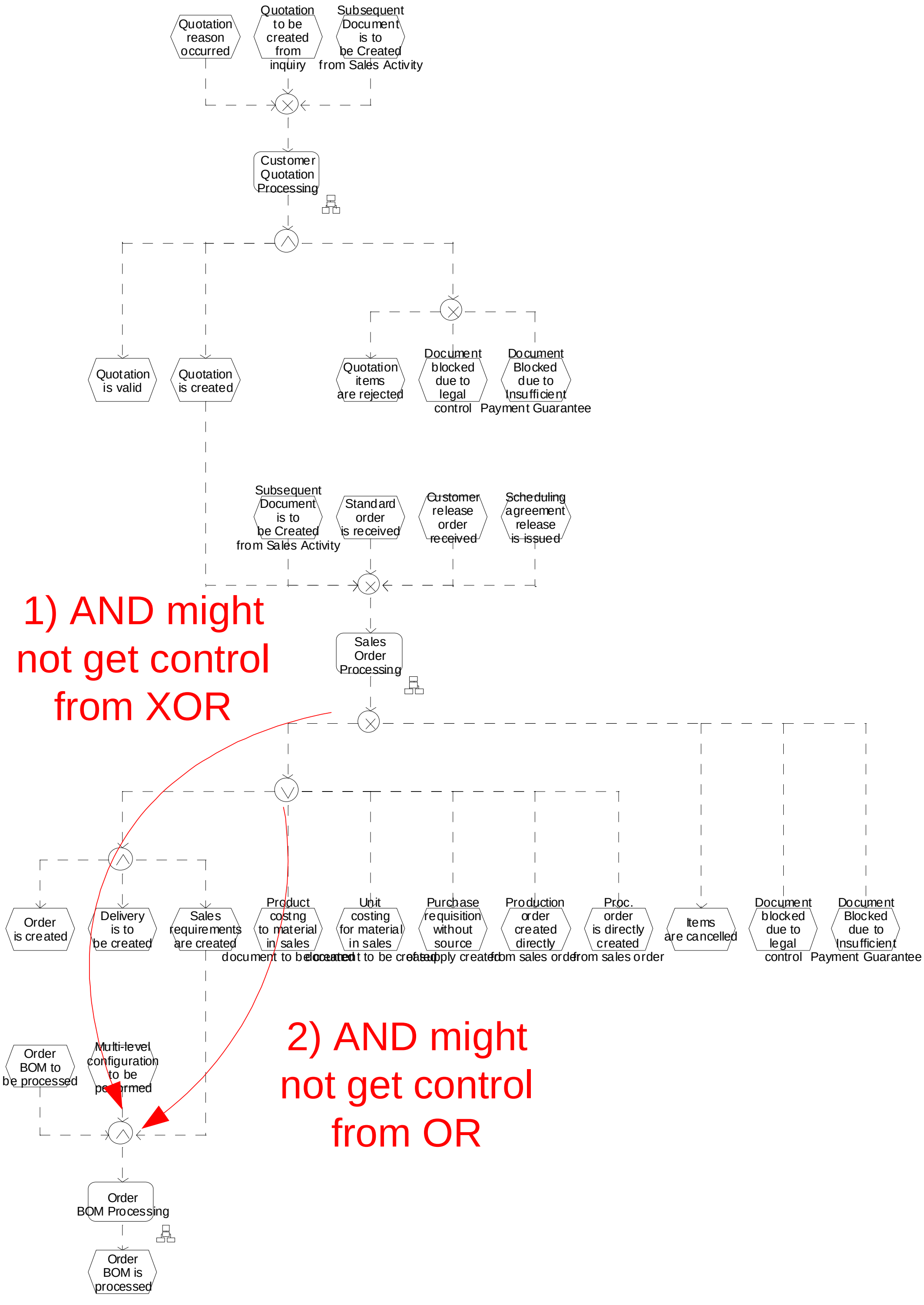


Figure A.77: Sales and Distribution – Sales Order Processing: Make/Assembly To Order – Sales order (completely reduced)

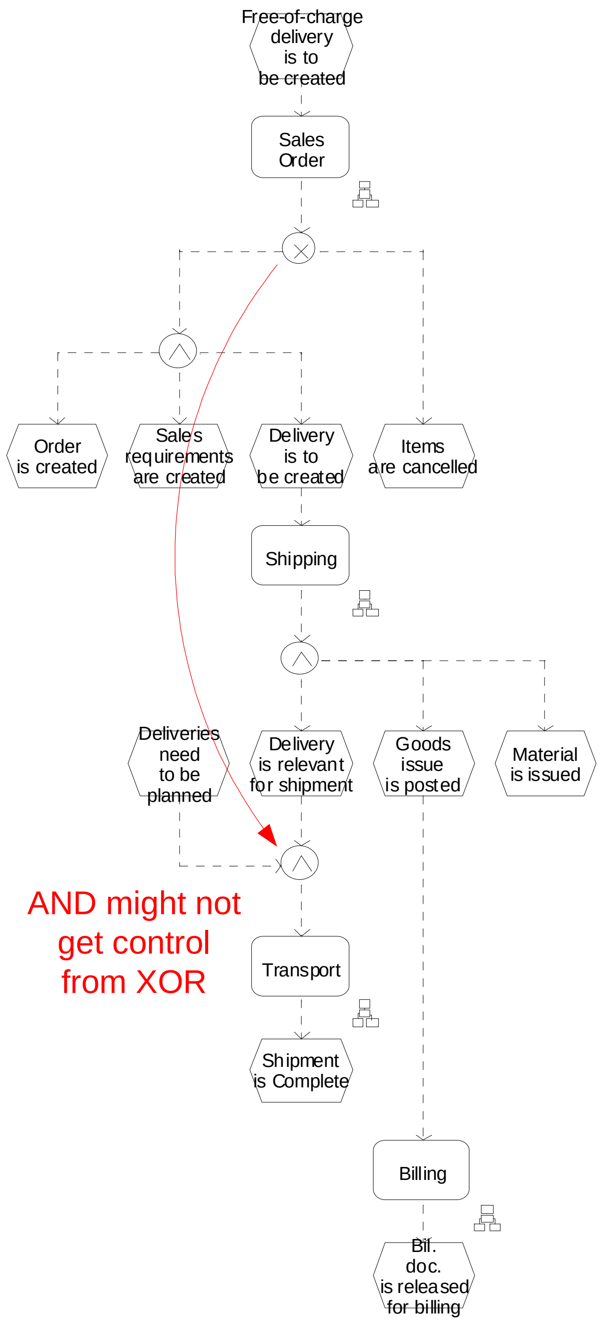


Figure A.78: Sales and Distribution – Sending Samples and Advertising Materials (completely reduced)

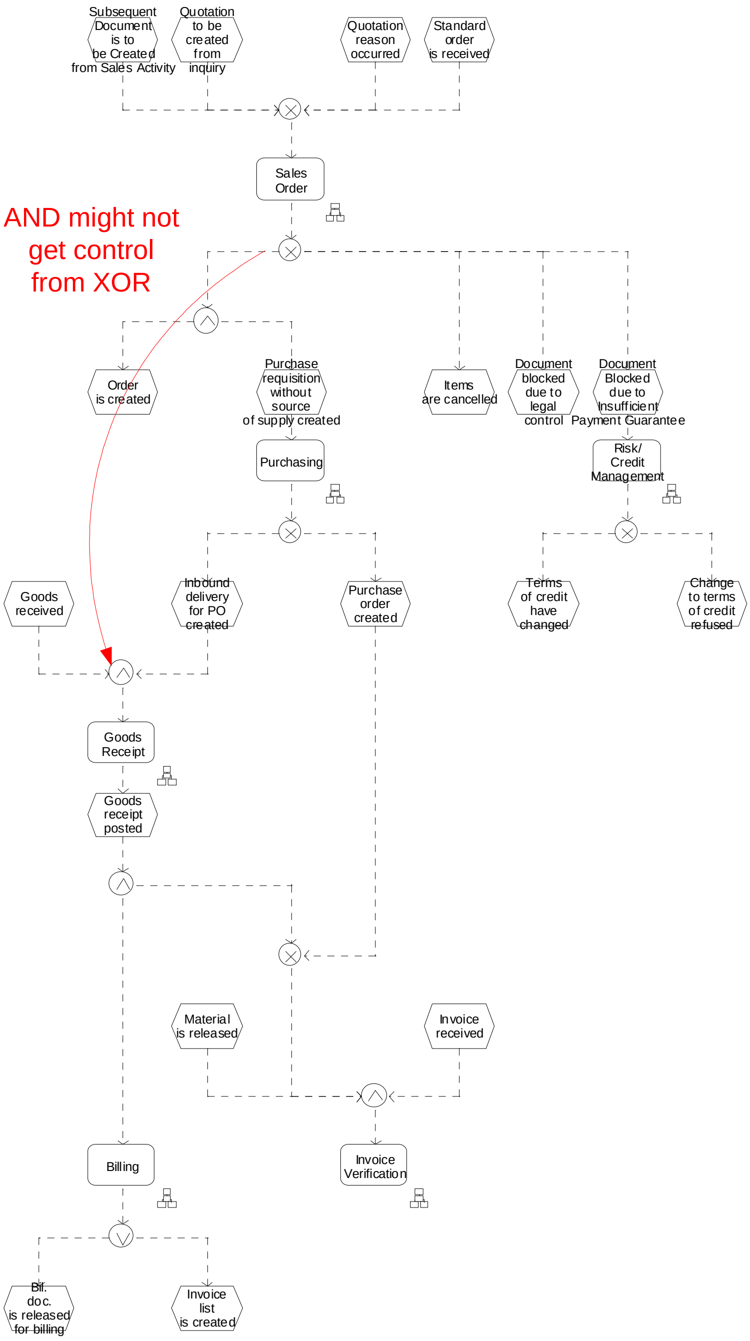


Figure A.79: Sales and Distribution – Third-Party Order Processing (reduced size 8)

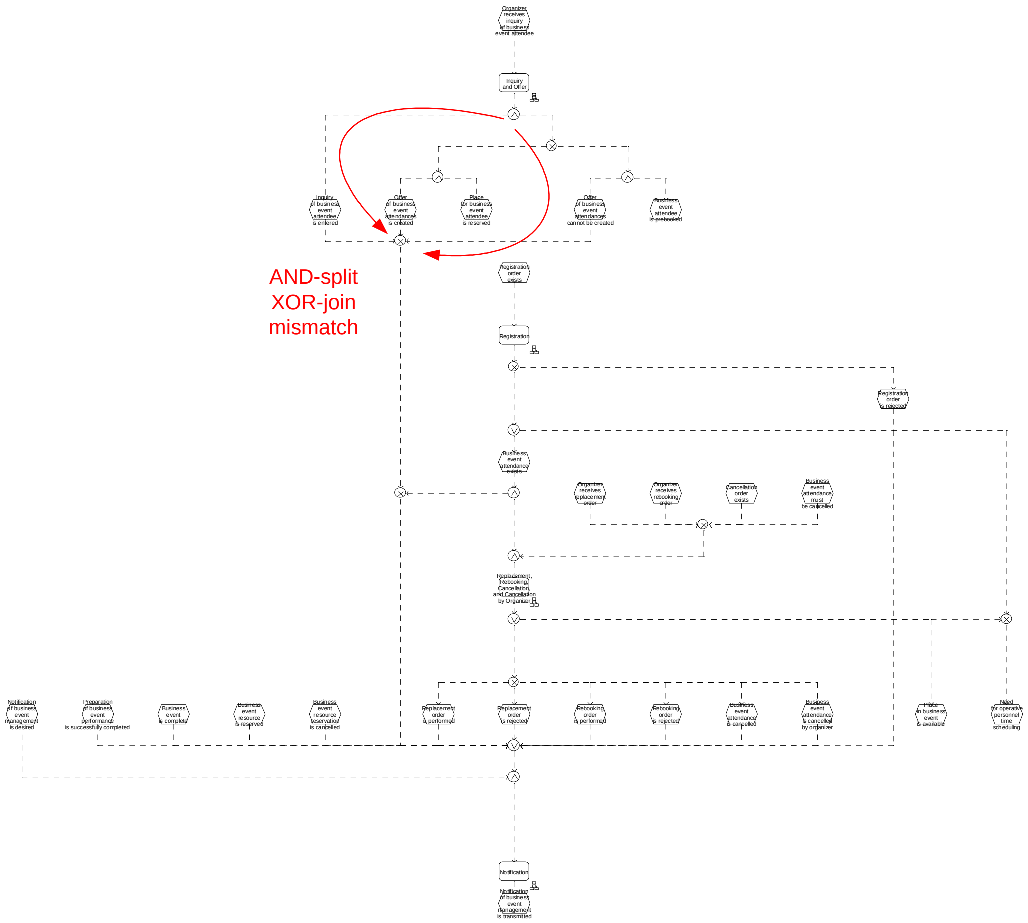


Figure A.80: Training and Event Management – Business Event Attendance Administration (reduced size 17)

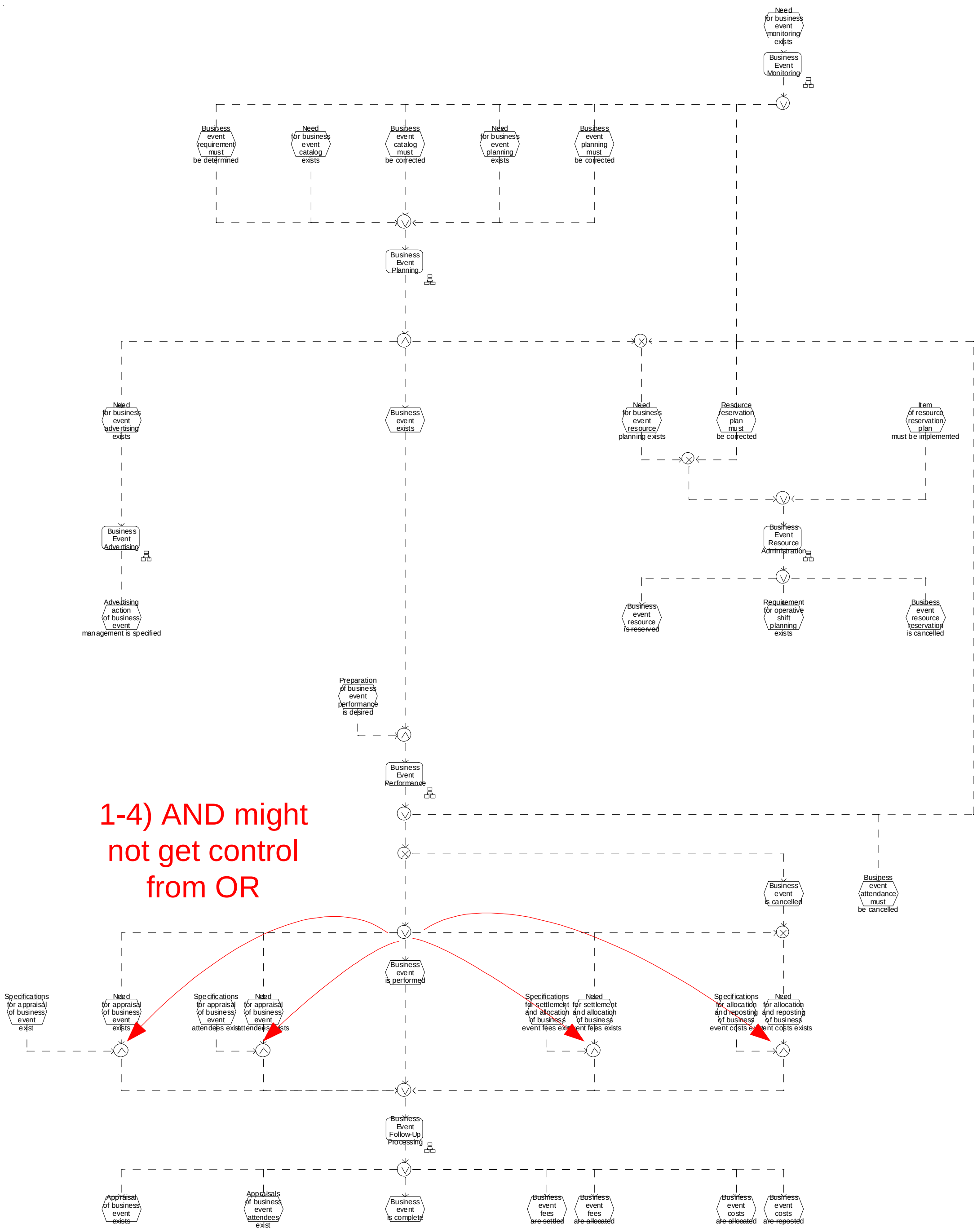


Figure A.81: Training and Event Management – Business Event Planning and Performance (reduced size 22)

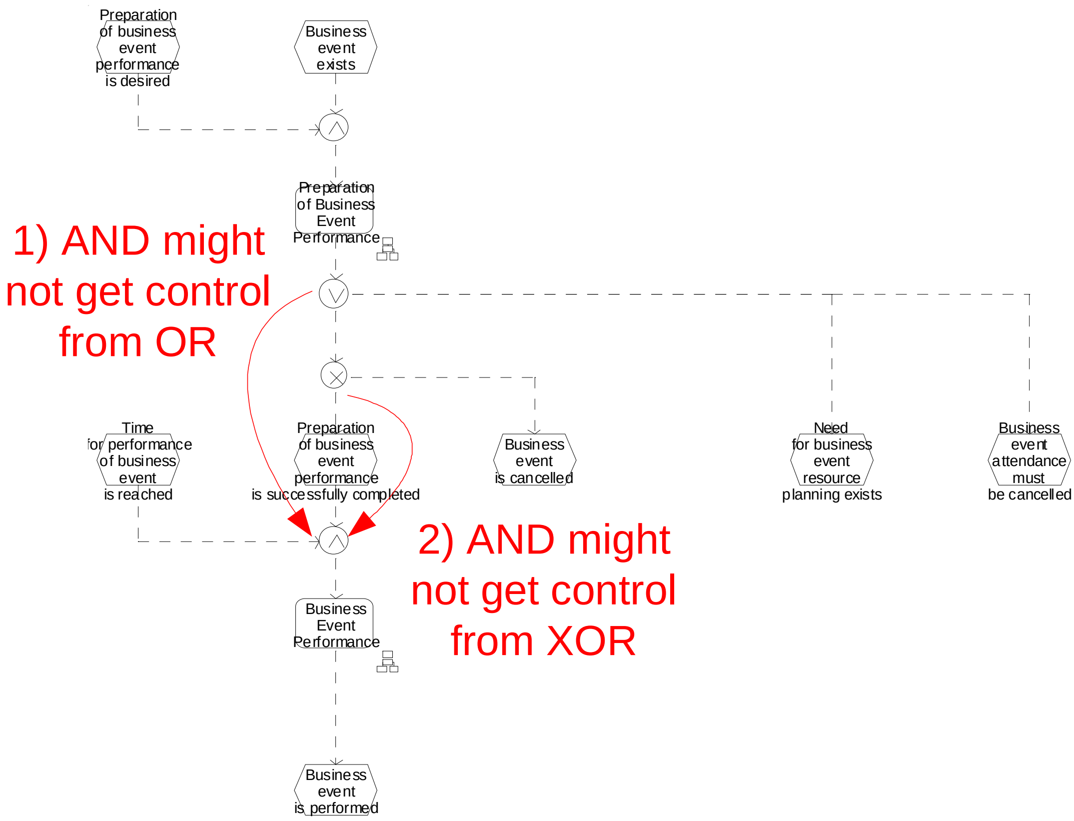


Figure A.82: Training and Event Management – Business Event Planning and Performance – Business Event Performance (completely reduced)

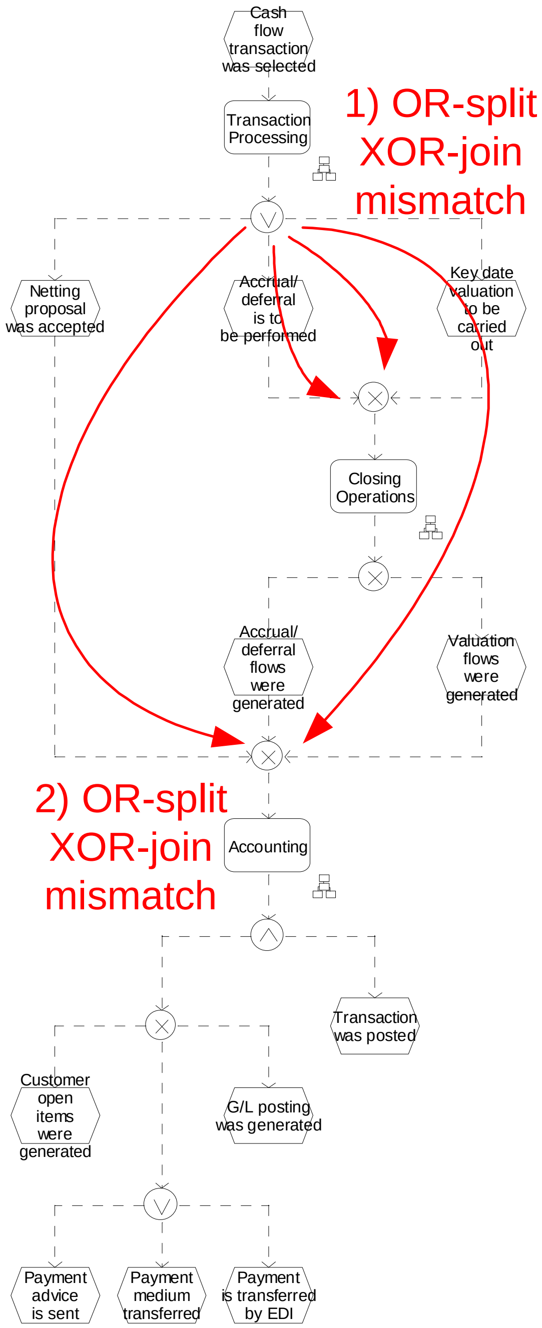


Figure A.83: Treasury – Cash Flow Transactions (TR-MM) (completely reduced)

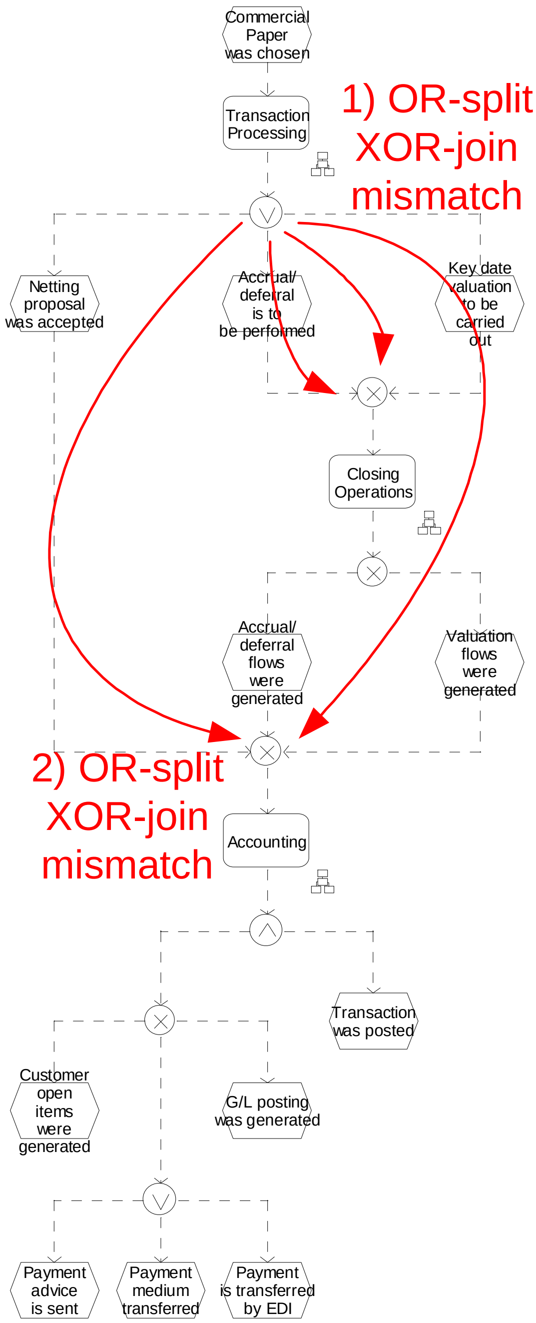


Figure A.84: Treasury – Commercial Paper (TR-MM) (completely reduced)

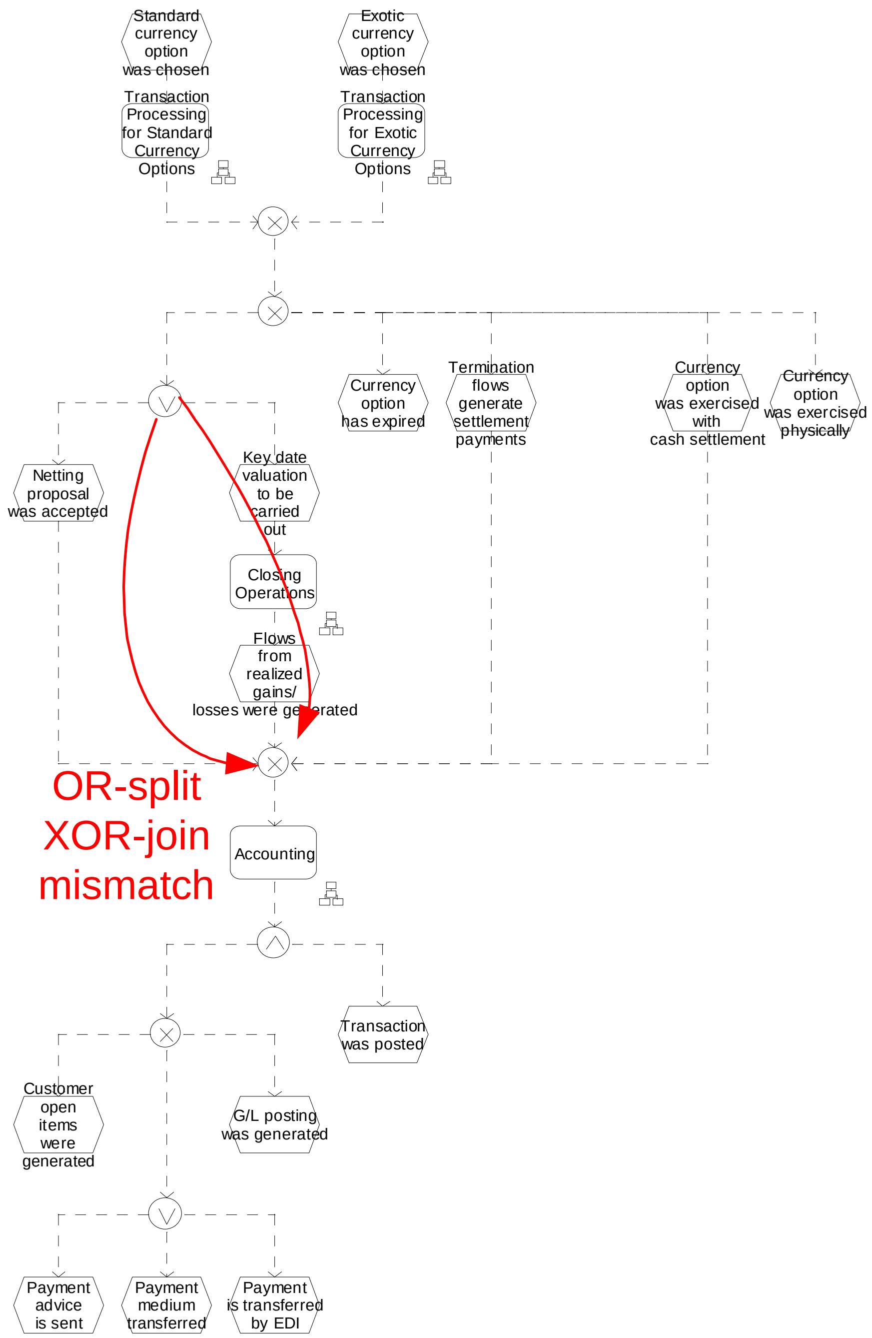


Figure A.85: Treasury – Currency Options (TR-FX) (completely reduced)

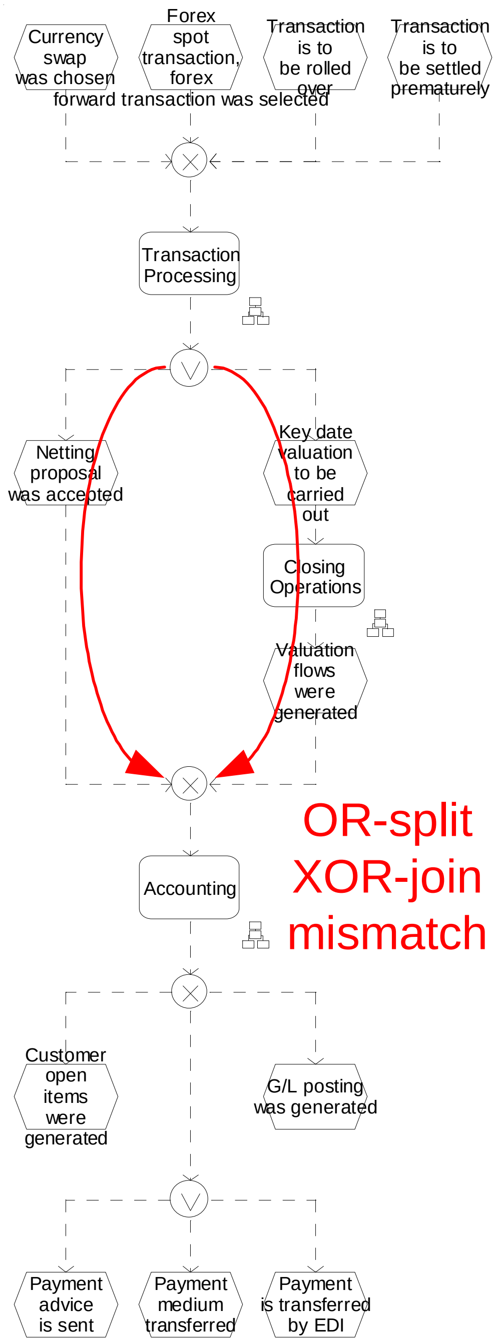


Figure A.86: Treasury – Forex Spot, Forward and Swap Transactions (TR-FX) (completely reduced)

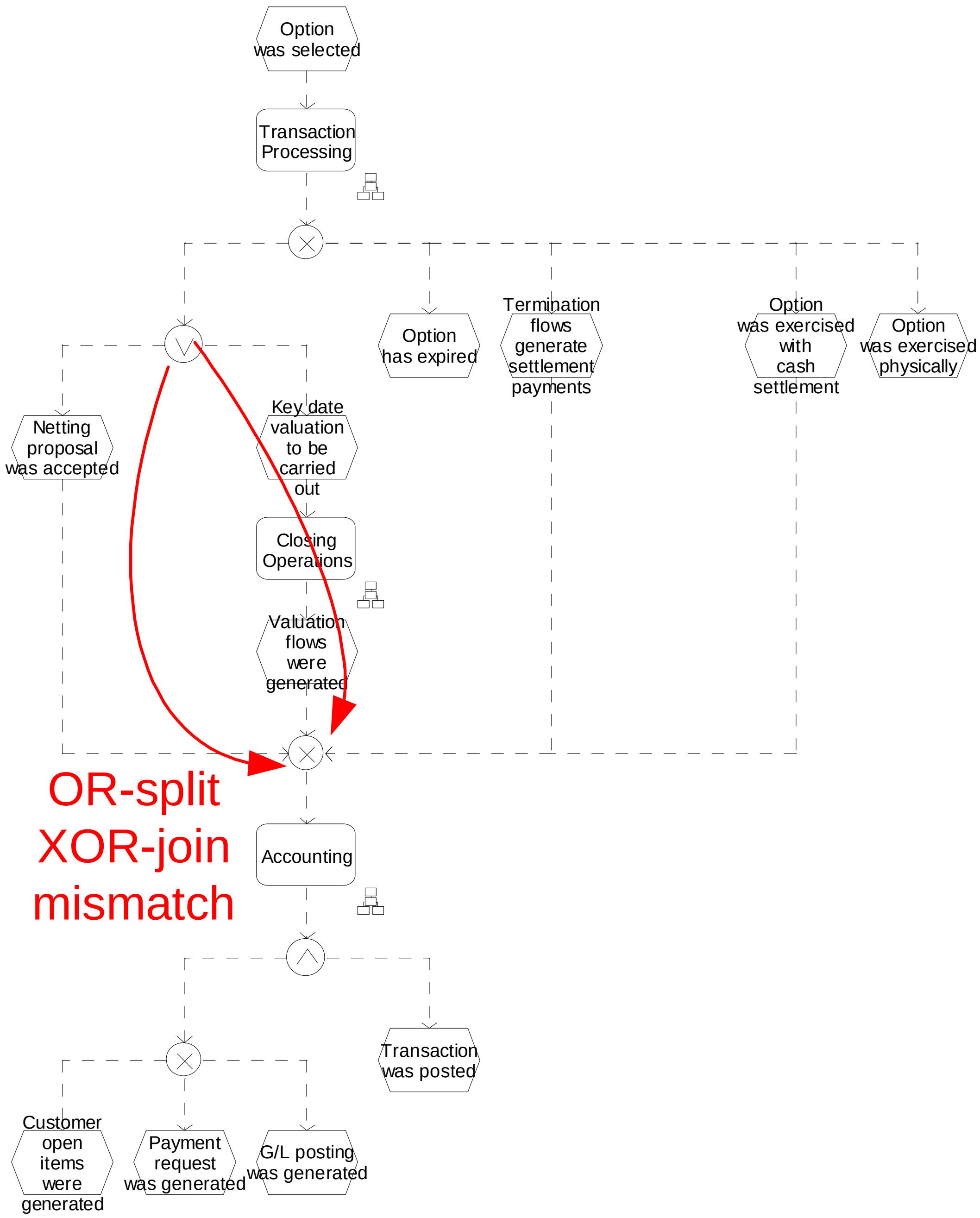


Figure A.87: Treasury – Options on Interest Rate Instruments and Securities (TR-DE) (completely reduced)

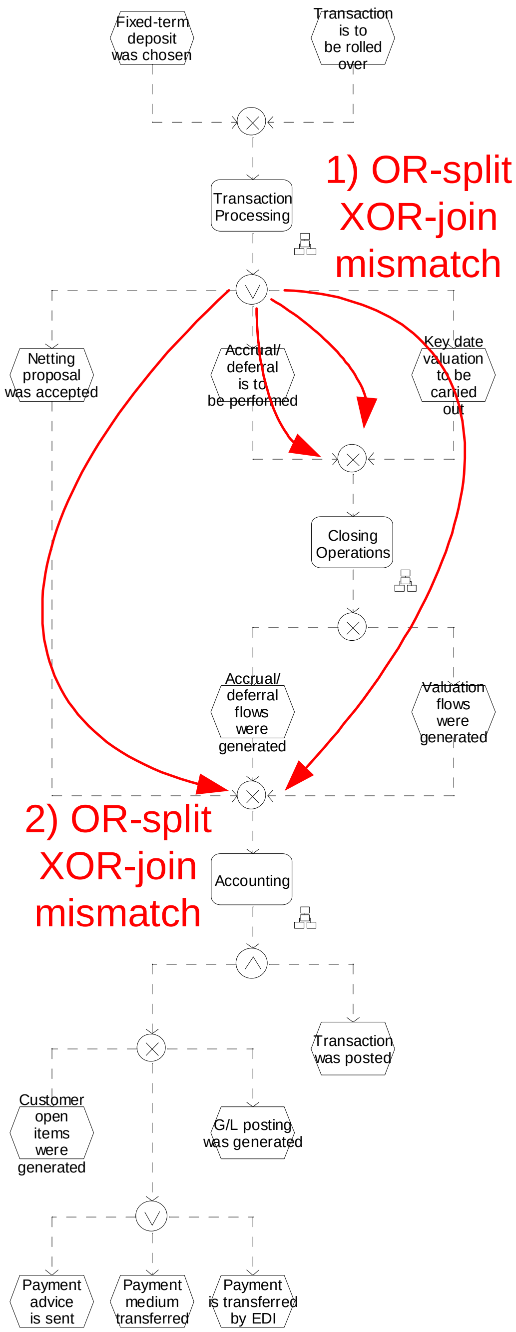


Figure A.88: Treasury – Process Fixed-Term Deposit (TR-MM) (completely reduced)

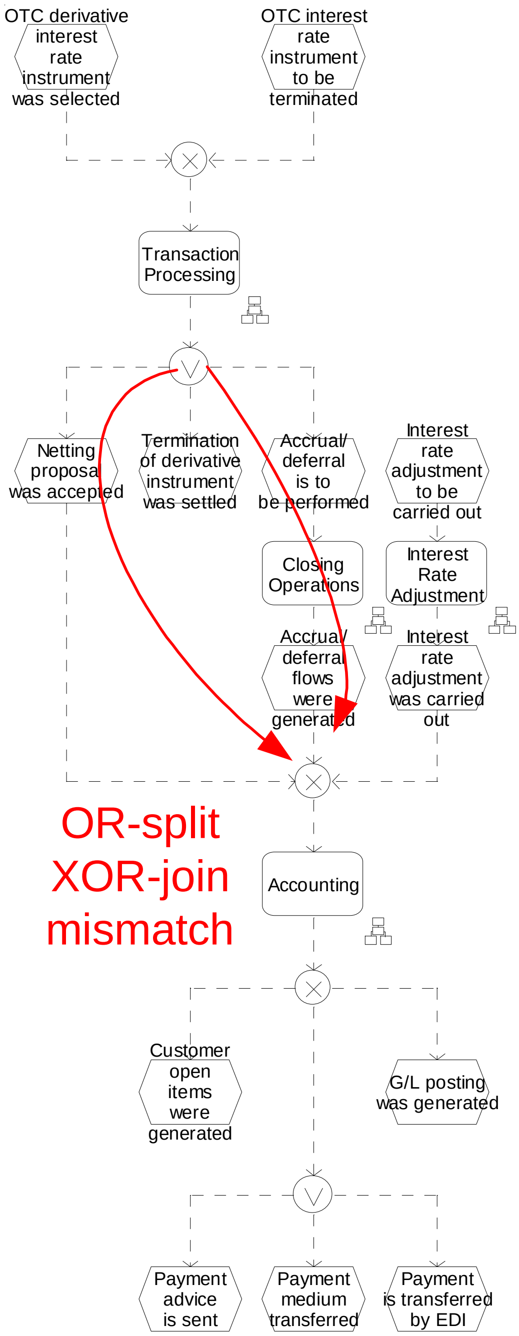


Figure A.89: Treasury – Process OTC Derivative Transactions (TR-DE) (reduced size 6)

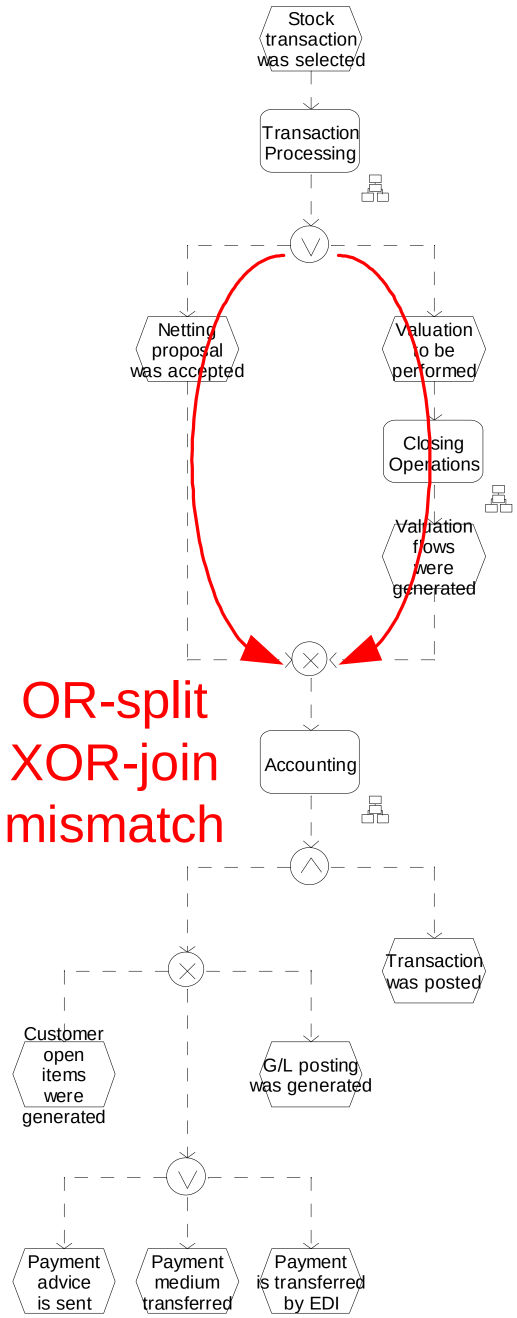


Figure A.90: Treasury – Stocks (TR-SE) (completely reduced)

Appendix B

EPCs not completely reduced

This appendix shows those EPCs of the SAP Reference Model that were not completely reduced and for which *xoEPC* did not find an error. We give the rest size in brackets and indicate whether ProM identified them to be sound or unsound.

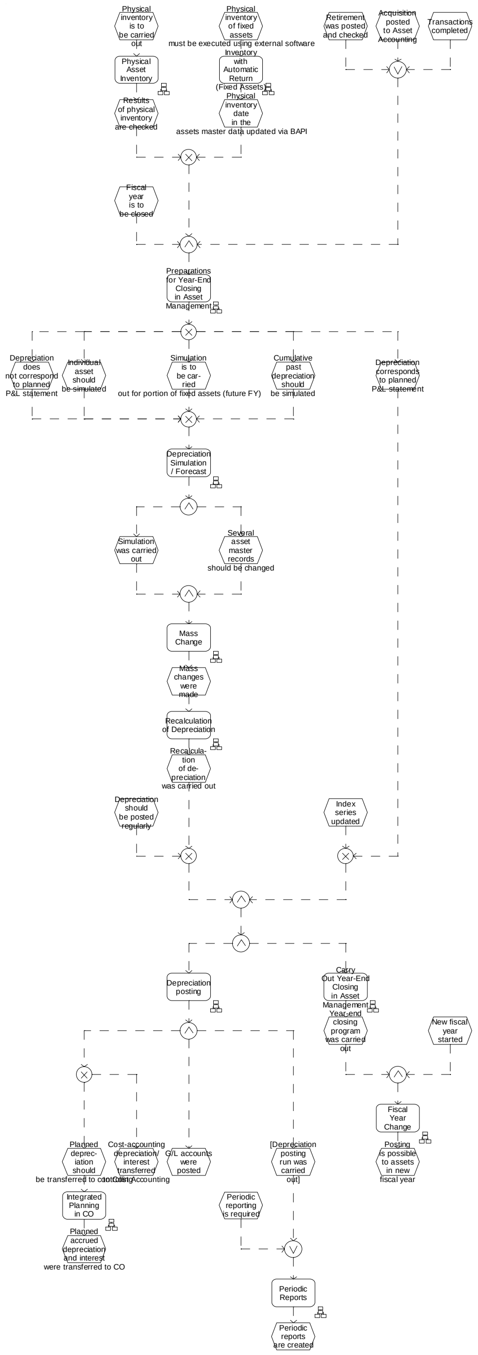


Figure B.1: Asset Accounting – Handling Fixed Assets – Closing Operations (Asset Accounting) (reduced size 14, unsound)

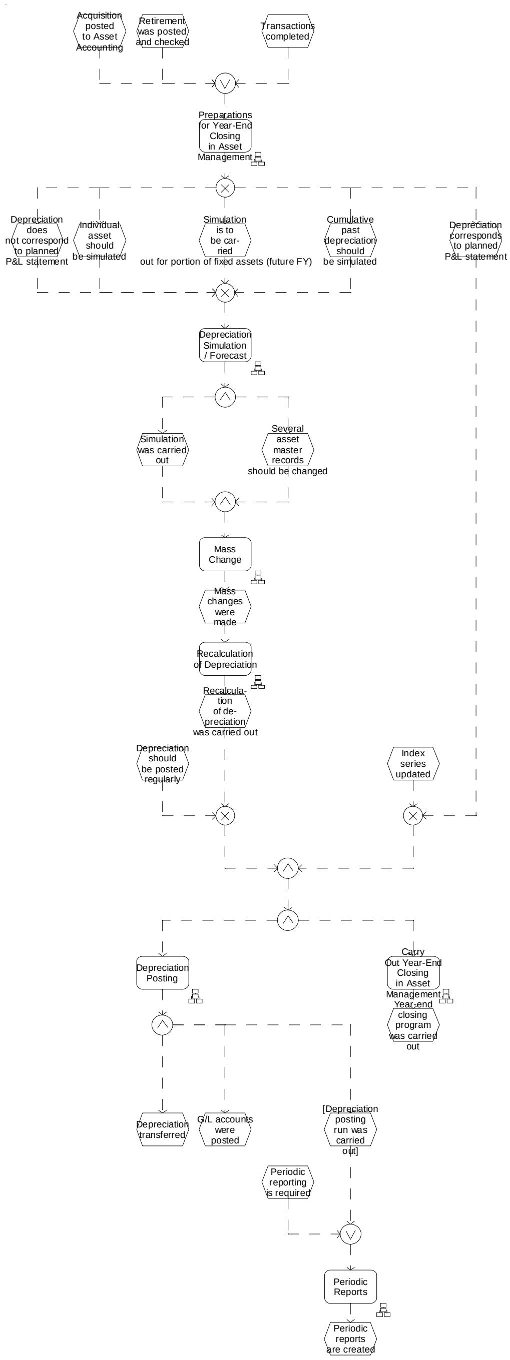


Figure B.2: Asset Accounting – Handling of Leased Assets – Closing Operations (reduced size 10, unsound)

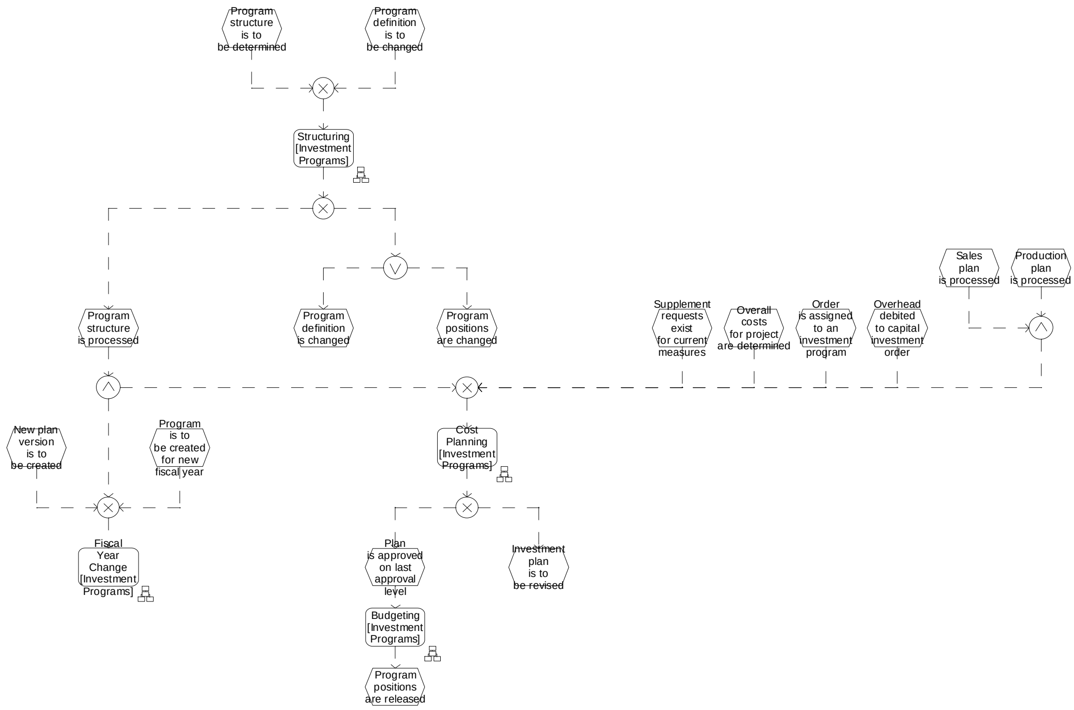


Figure B.3: Asset Accounting – Investment Program Handling (Capital Investments) (reduced size 10, sound)

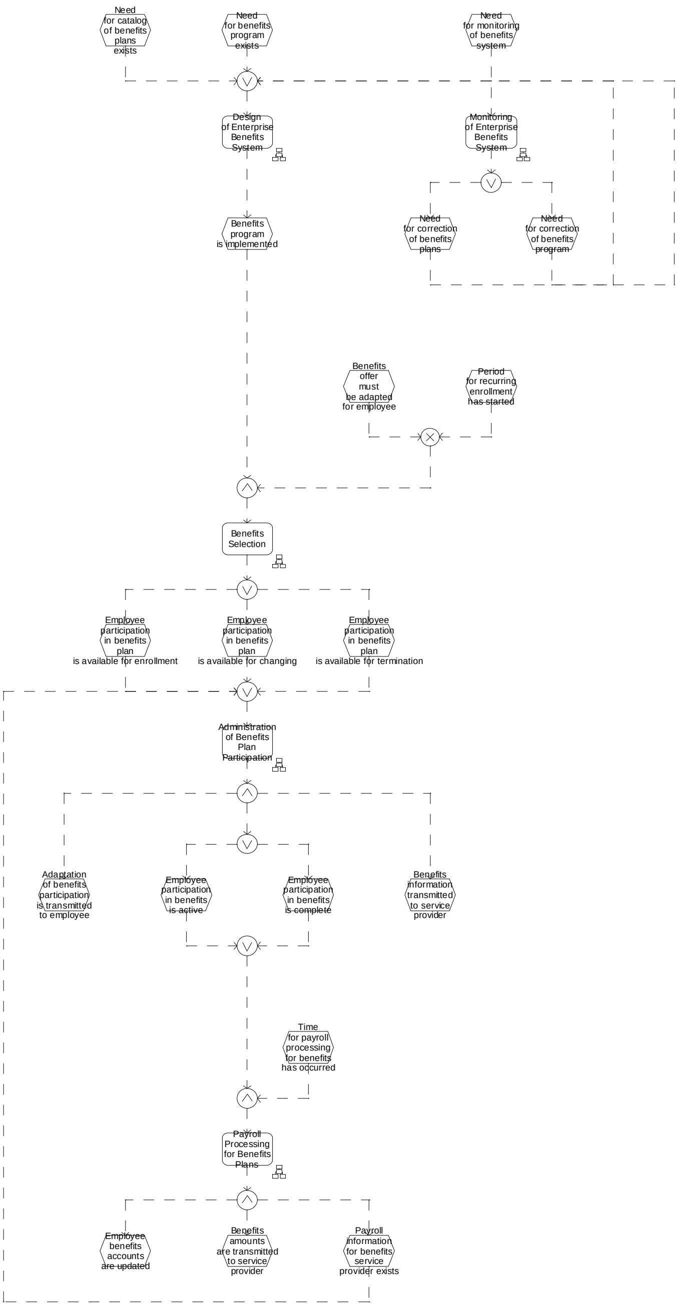


Figure B.4: Benefits Administration – Benefits Administration (reduced size 8, unsound)

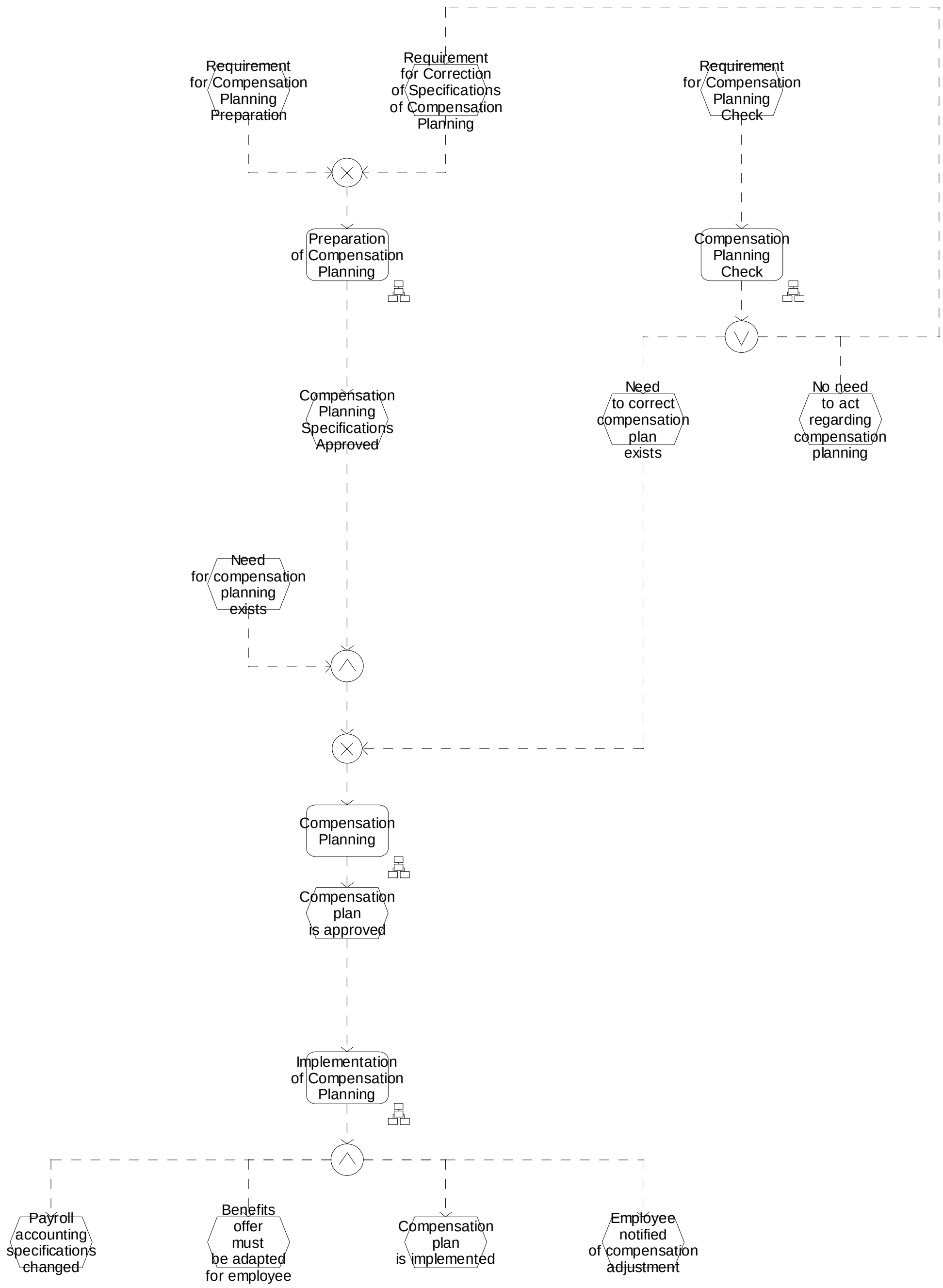


Figure B.5: Compensation Management – Compensation Planning (reduced size 9, unsound)

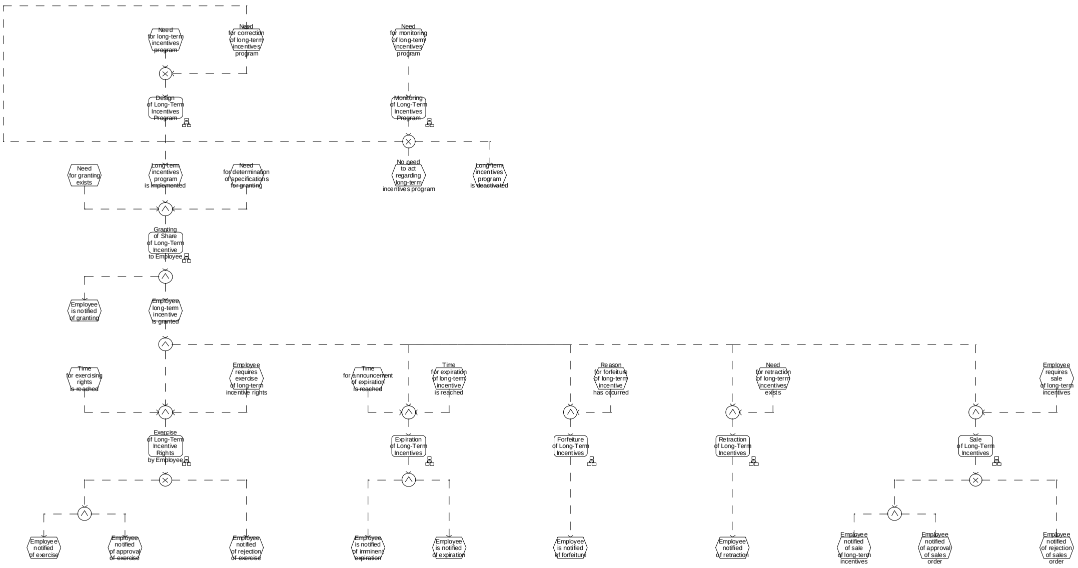


Figure B.6: Compensation Management – Long-Term Incentives (reduced size 23, unsound)

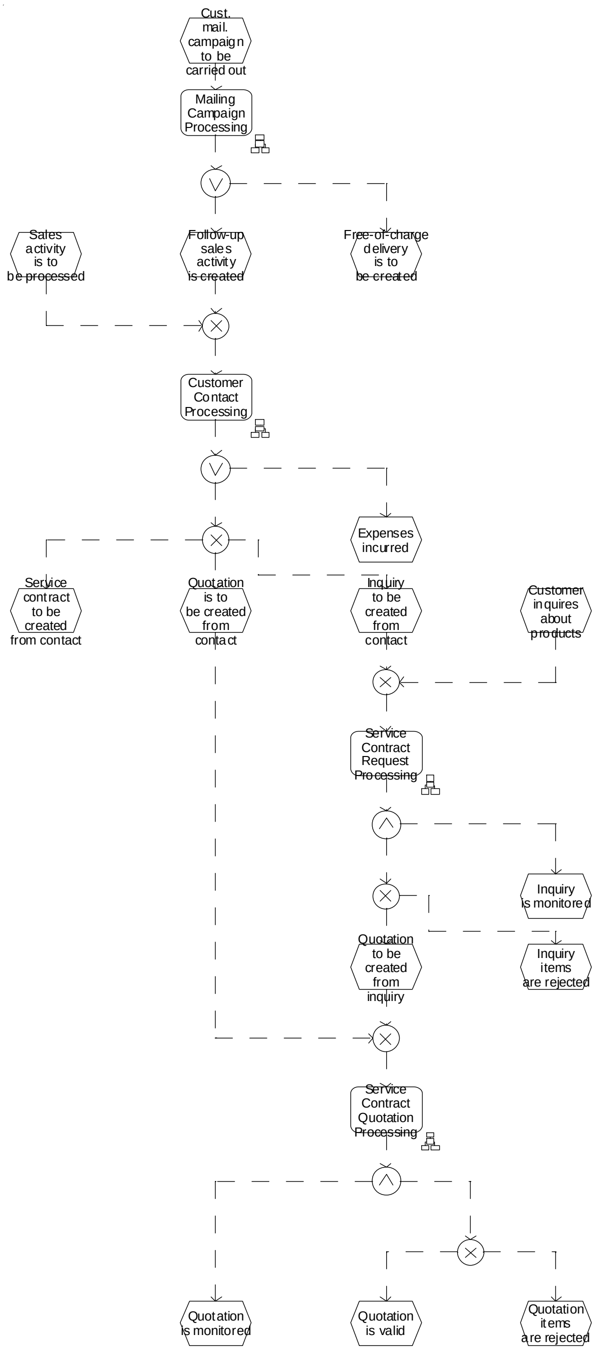


Figure B.7: Customer Service – Long-Term Service Agreements – Presales Activities (reduced size 15, sound)

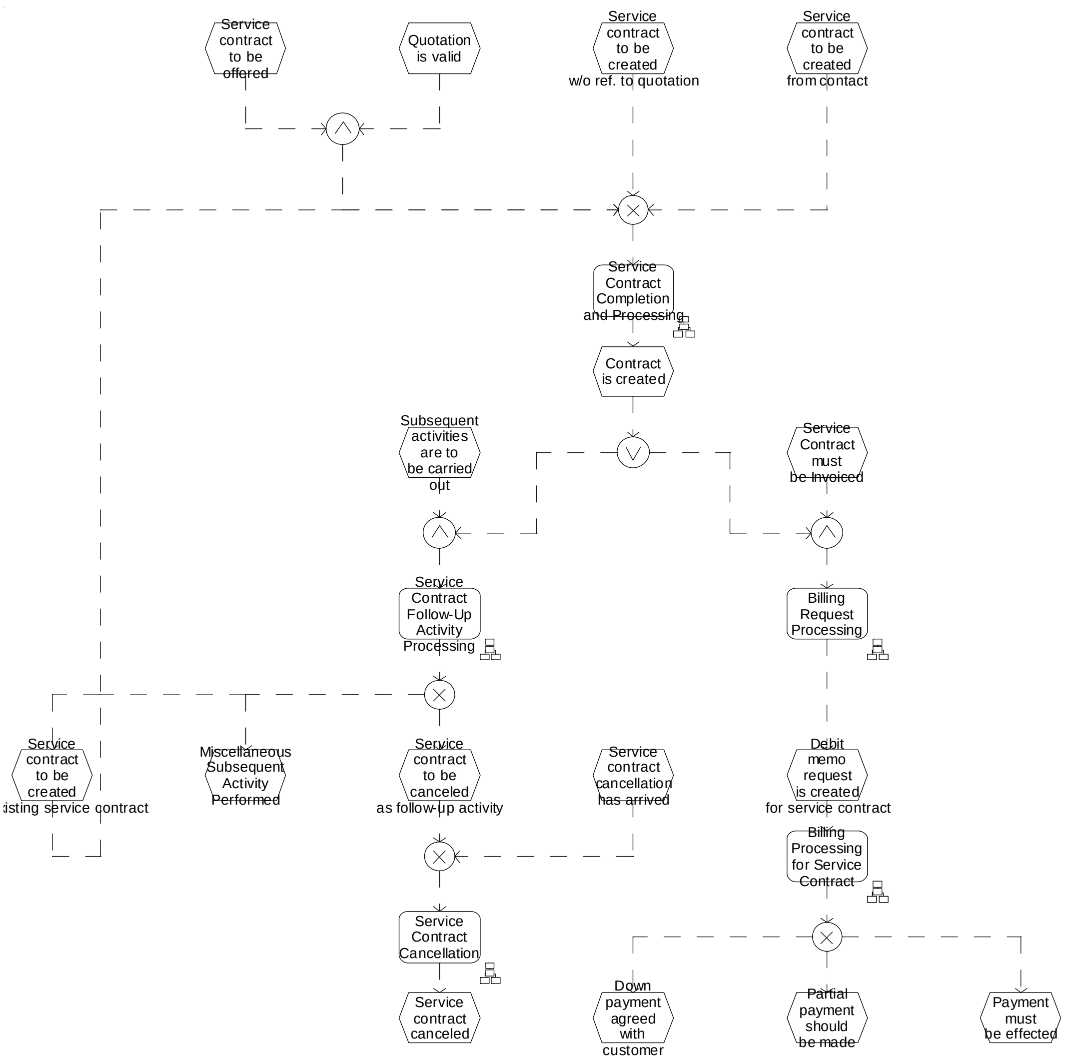


Figure B.8: Customer Service – Long-Term Service Agreements – Service Contract Processing (reduced size 13, unsound)

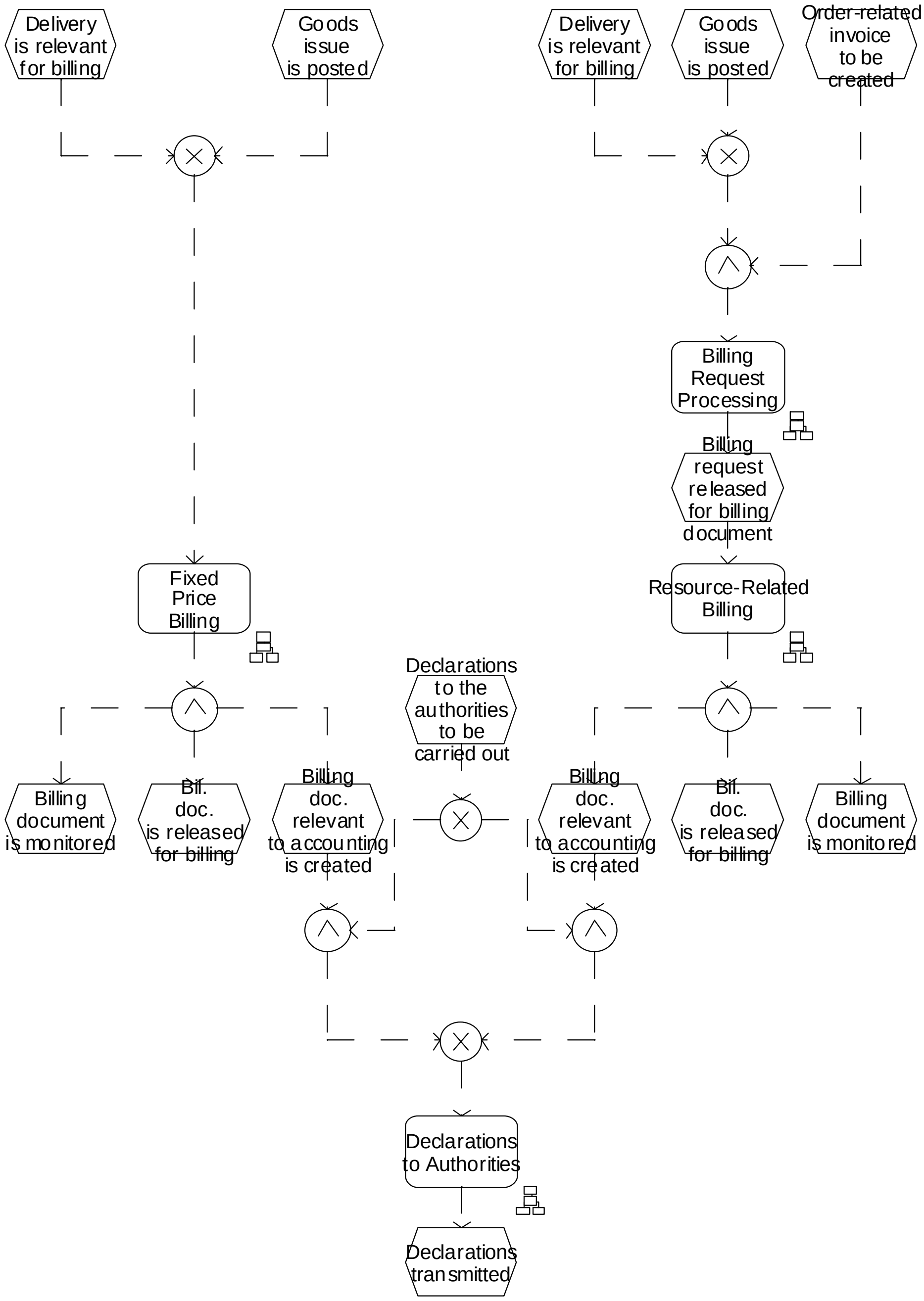


Figure B.9: Customer Service – Repairs Processing at Customer (Field Service) – Billing (reduced size 8, unsound)

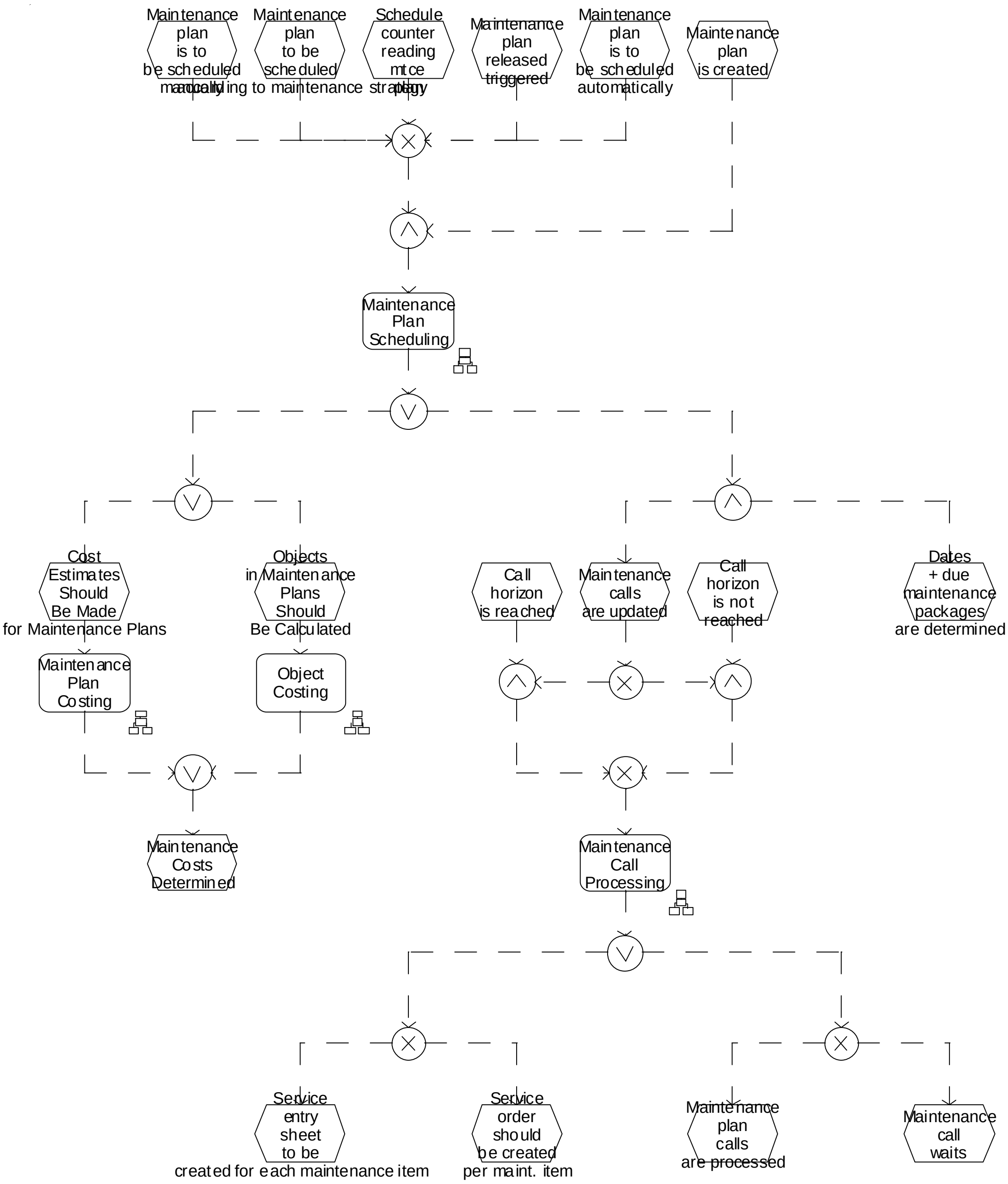


Figure B.10: Customer Service – Repairs Processing at Customer (Field Service) – Maintenance Planning (reduced size 10, unsound)

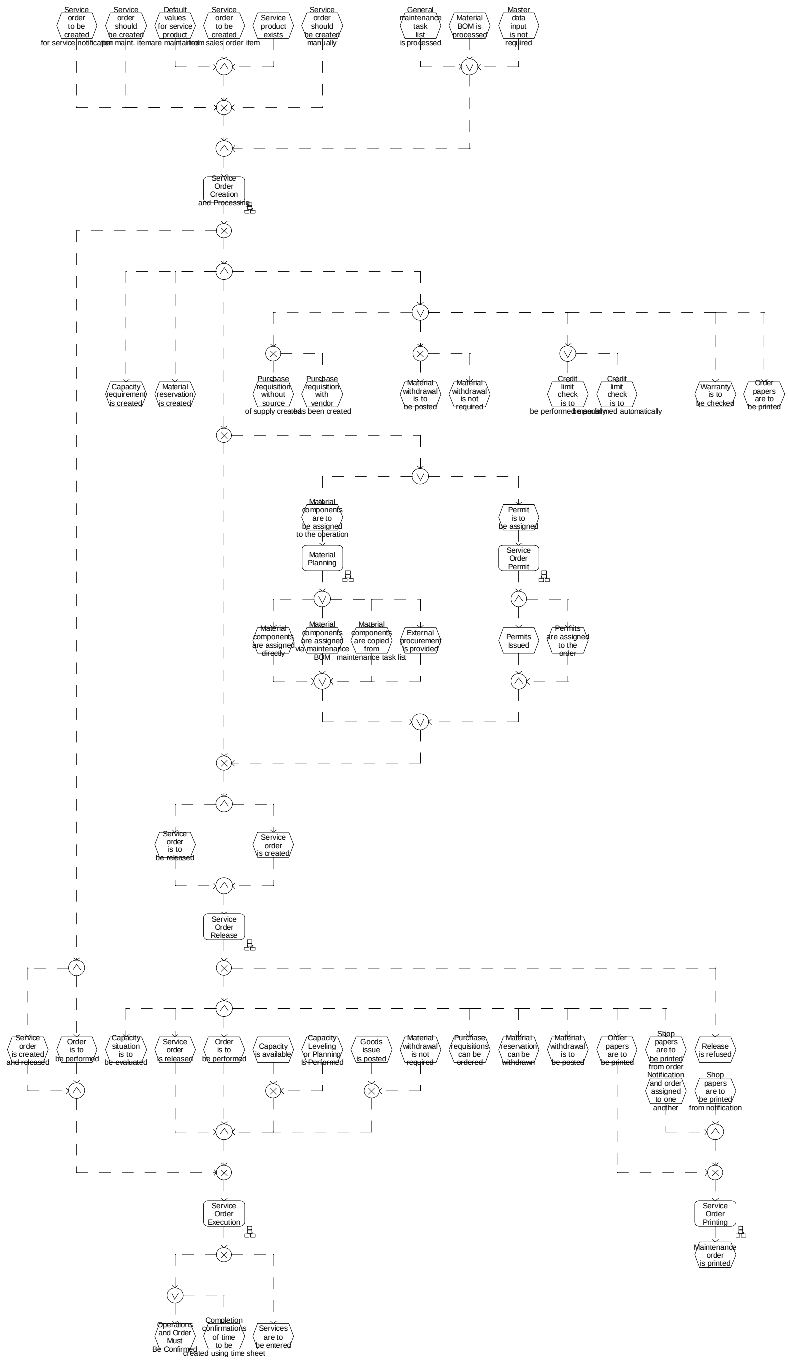


Figure B.11: Customer Service – Repairs Processing at Customer (Field Service) – Service Order (reduced size 11, unsound)

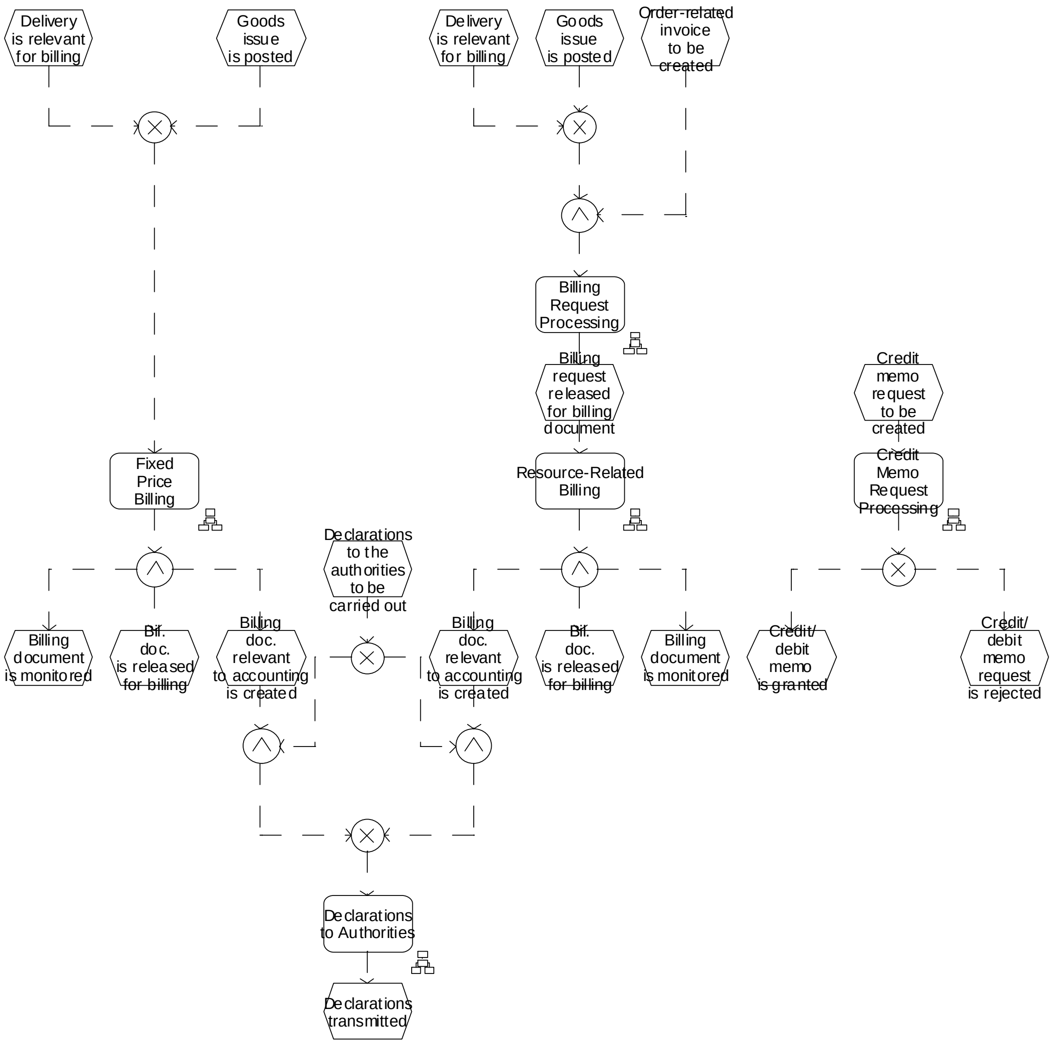


Figure B.12: Customer Service – Repairs Processing in Service Center (Inhouse) – Billing (reduced size 8, unsound)

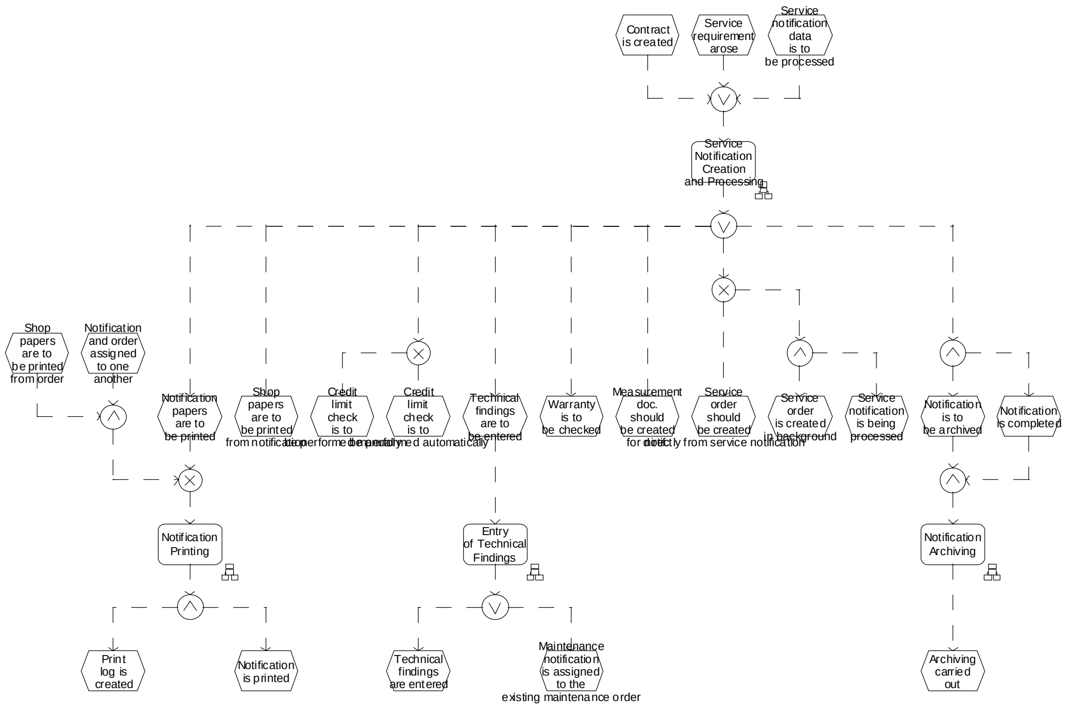


Figure B.13: Customer Service – Repairs Processing in Service Center (Inhouse) – Service Notification (reduced size 6, sound)

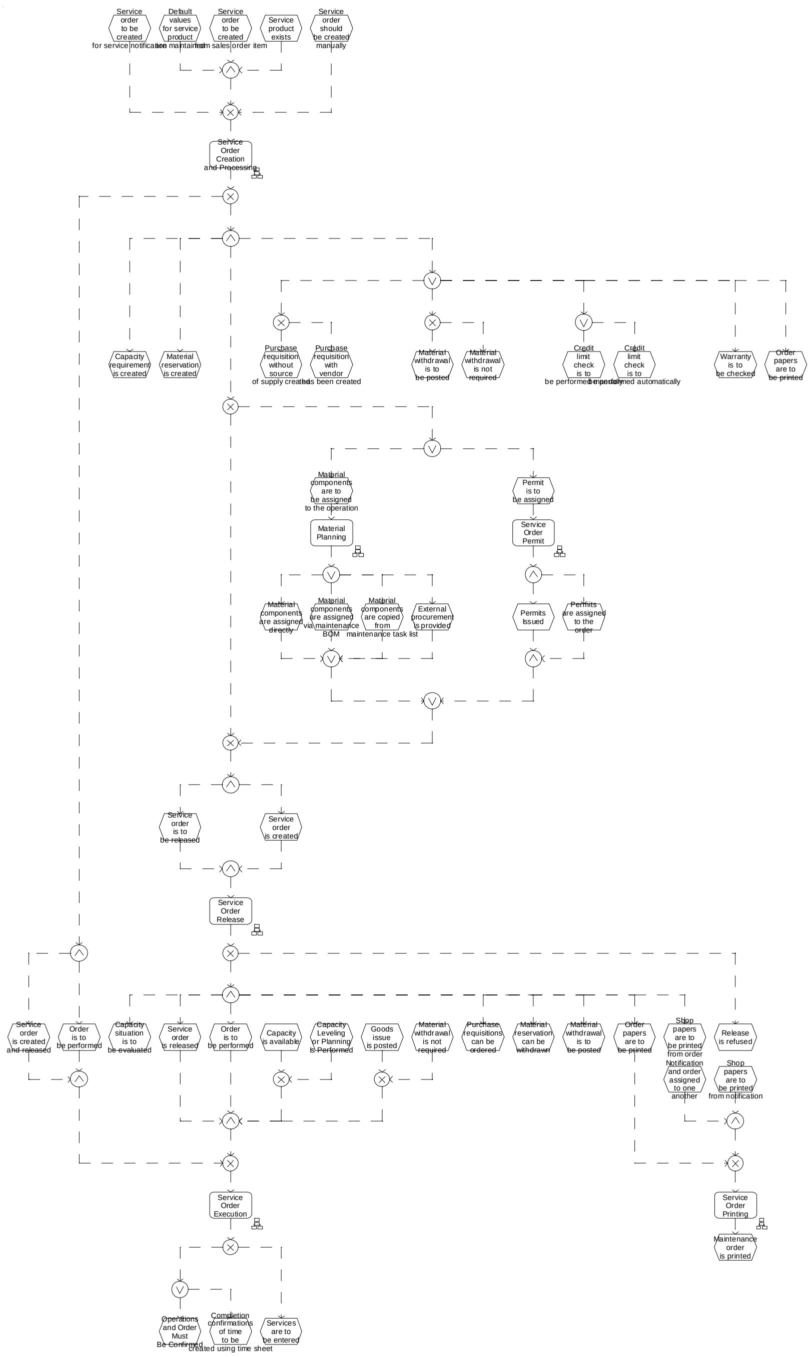


Figure B.14: Customer Service – Repairs Processing in Service Center (Inhouse) – Service Order (reduced size 11, unsound)

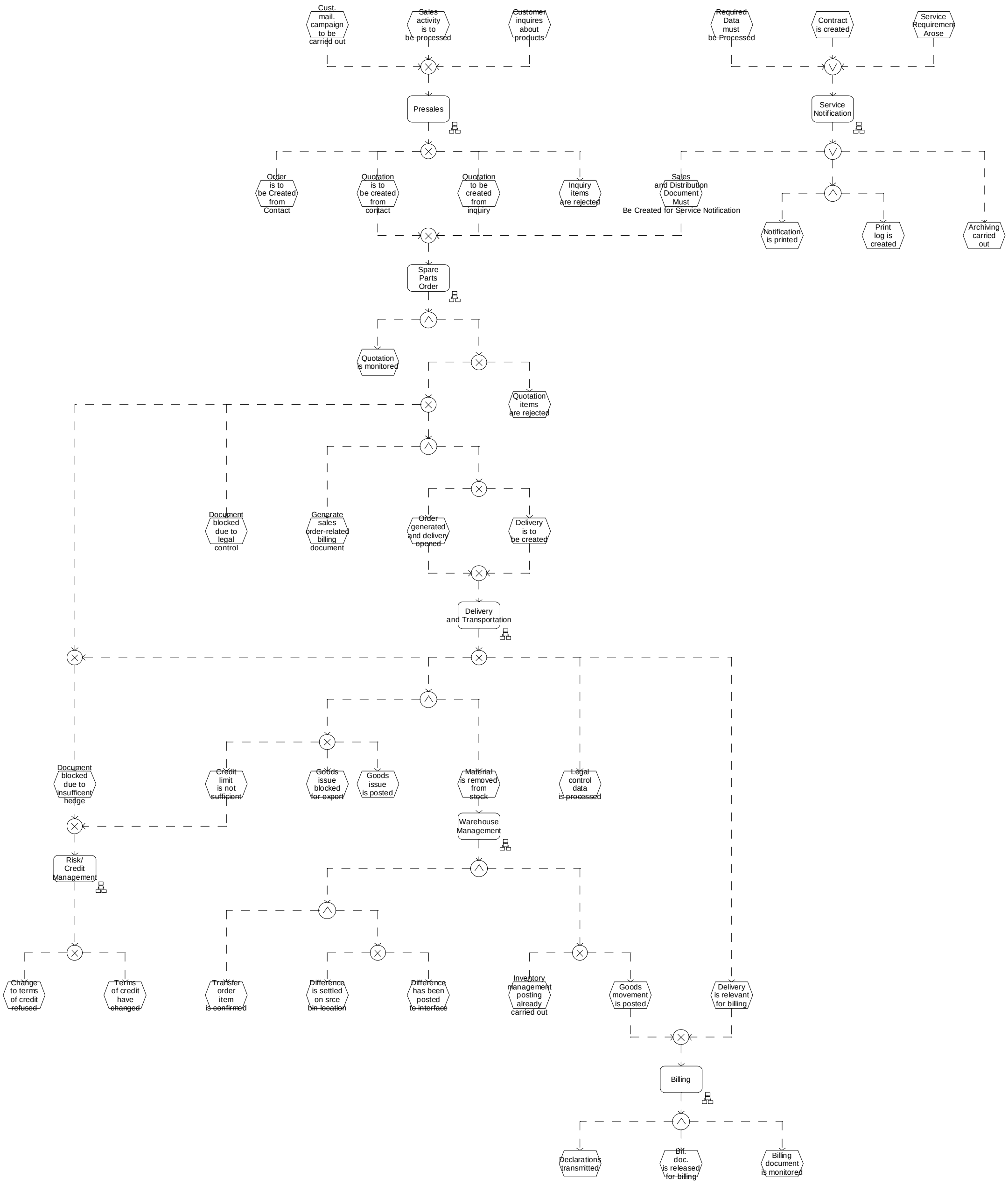


Figure B.15: Customer Service – Spare Parts Delivery Processing (reduced size 18, sound)

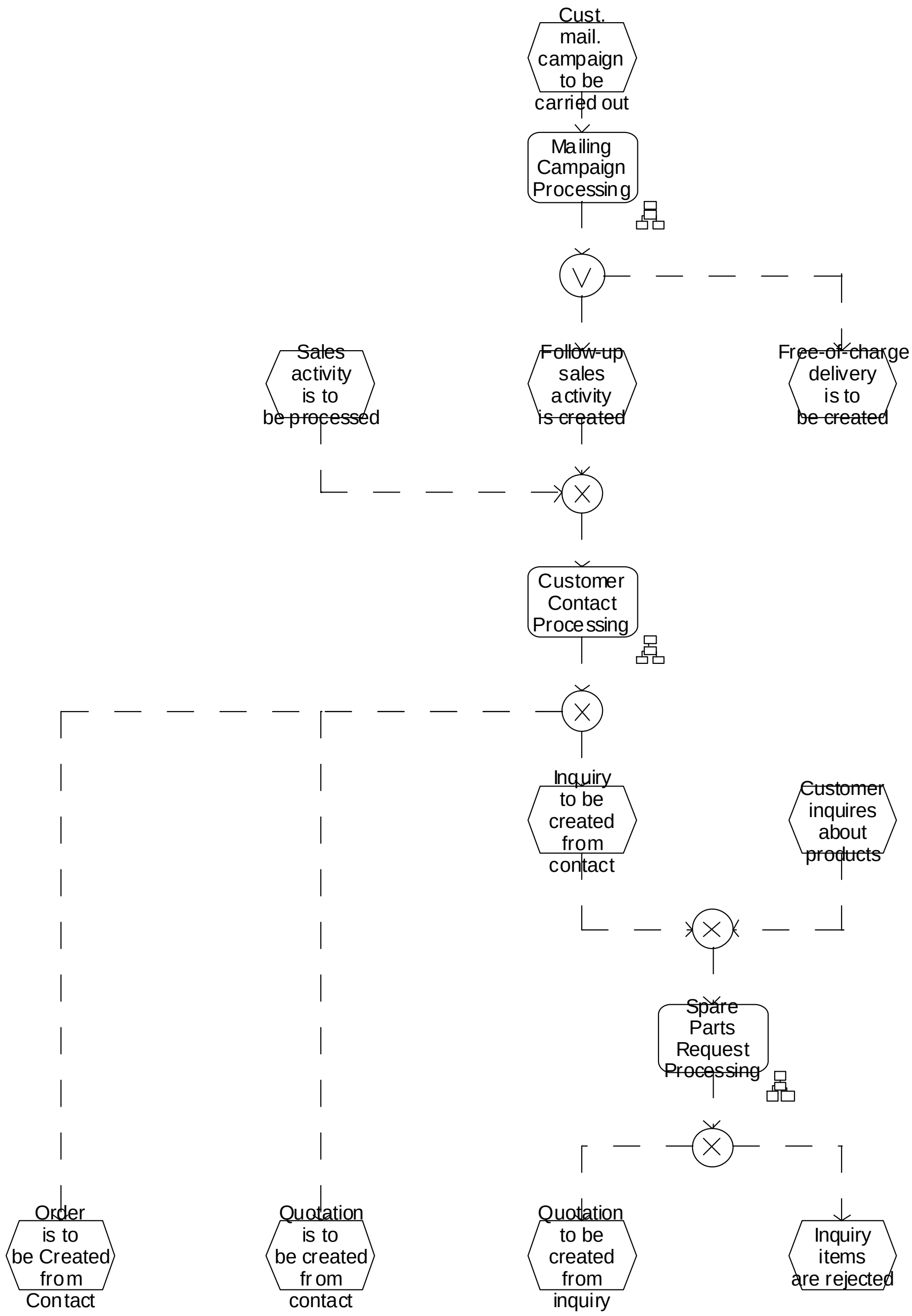


Figure B.16: Customer Service – Spare Parts Delivery Processing – Presales (reduced size 10, sound)

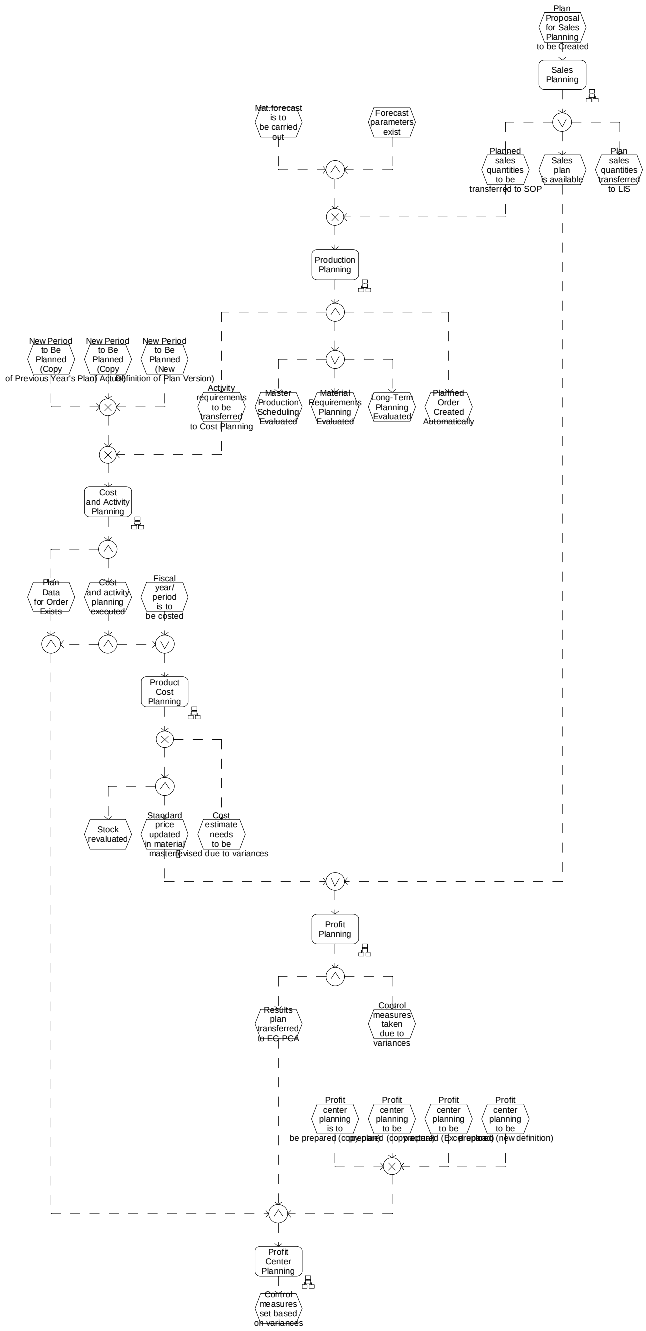


Figure B.17: Enterprise Controlling – Operational business planning (reduced size 14, unsound)

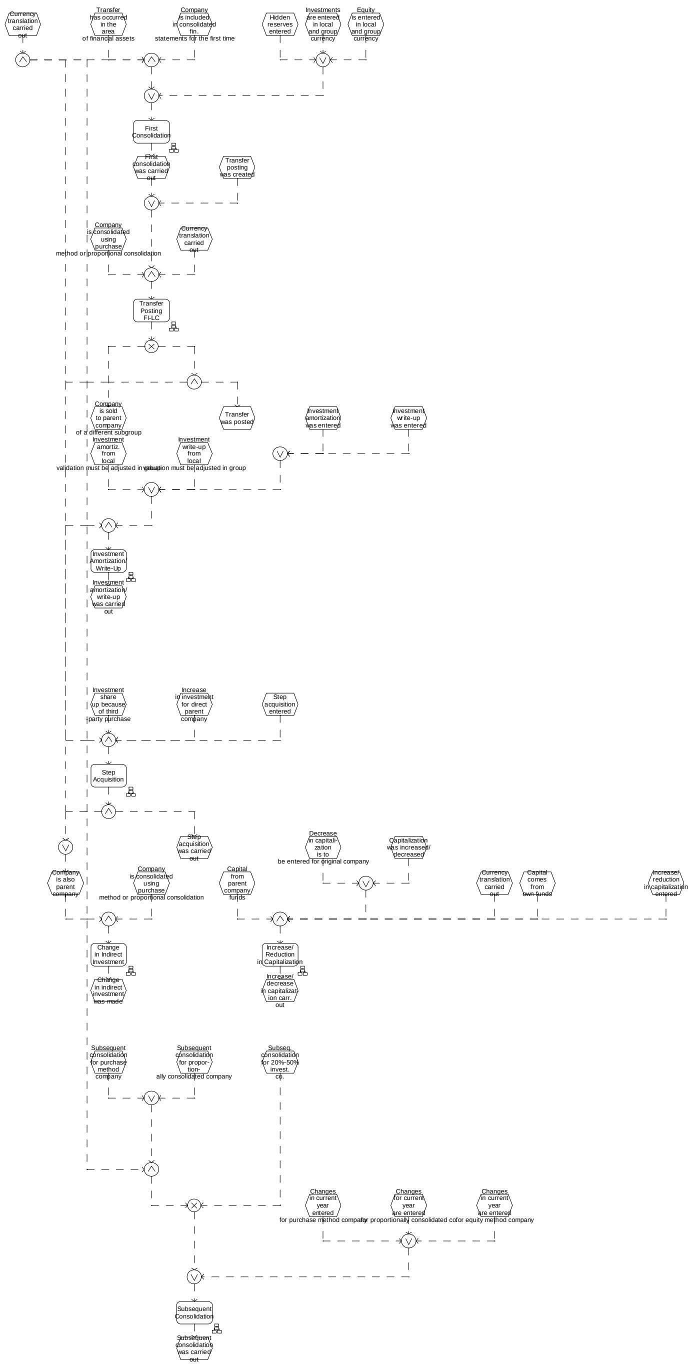


Figure B.18: Financial Accounting – Consolidation – Consolidation of Investments (reduced size 26, sound)

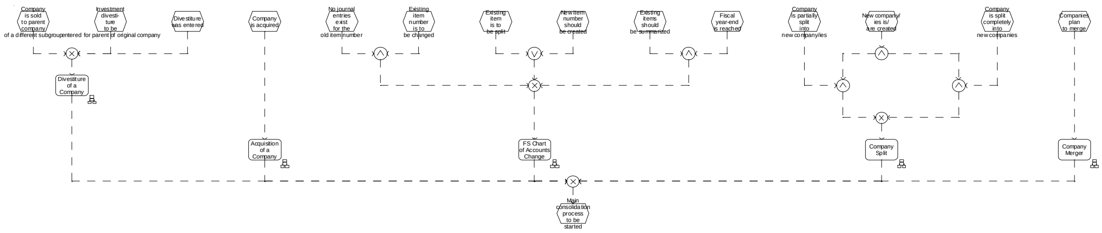


Figure B.19: Financial Accounting – Consolidation – Master Data Maintenance (reduced size 9, unsound)

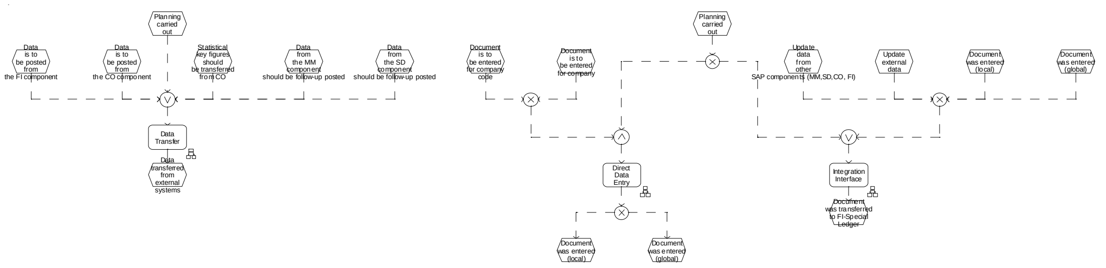


Figure B.20: Financial Accounting – Special Purpose Ledger – Actual Posting (reduced size 8, unsound)

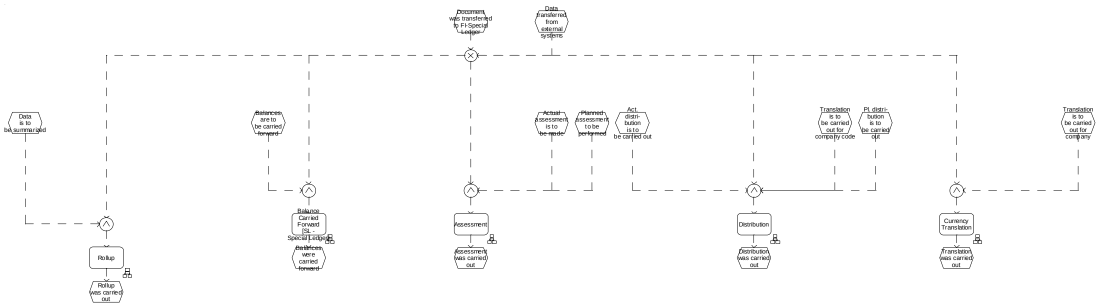


Figure B.21: Financial Accounting – Special Purpose Ledger – Periodic Processing (reduced size 17, unsound)

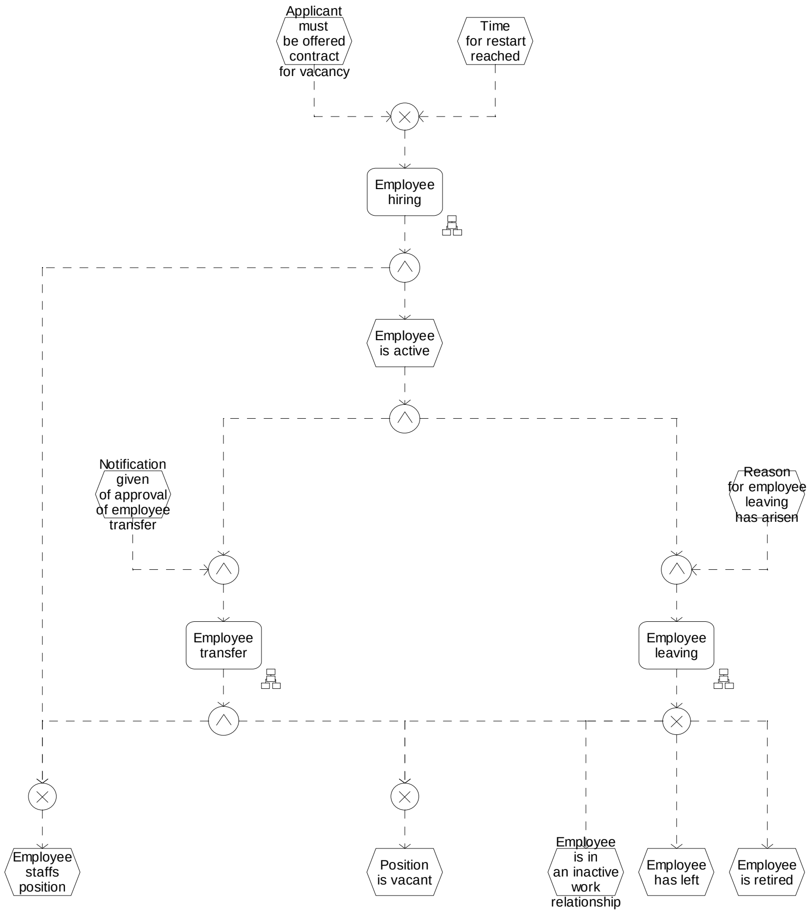


Figure B.22: Personnel Administration – Personnel Actions (reduced size 13, unsound)

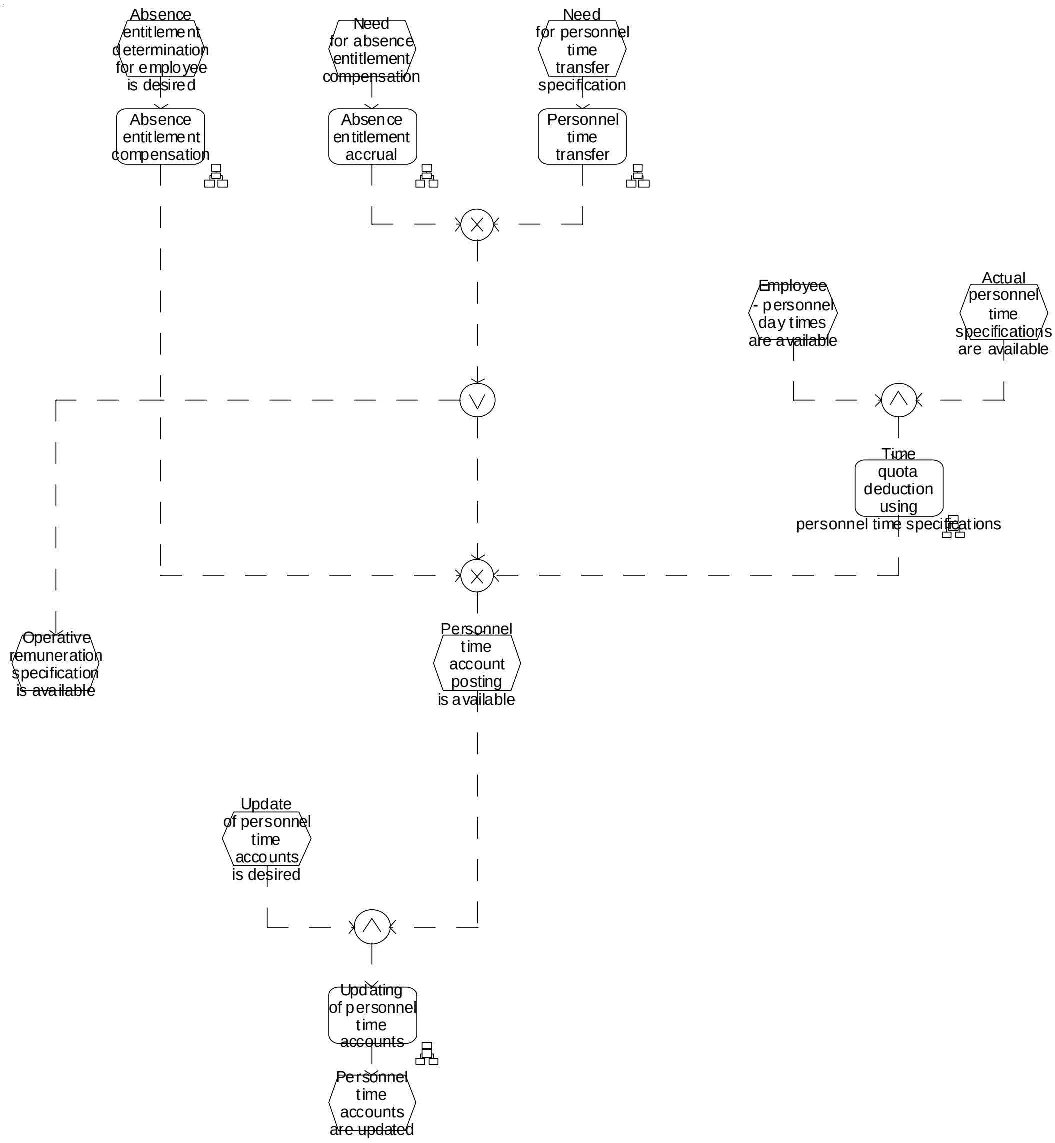


Figure B.23: Personnel Time Management – Personnel Time Management – Personnel time accounts administration (reduced size 8, unsound)

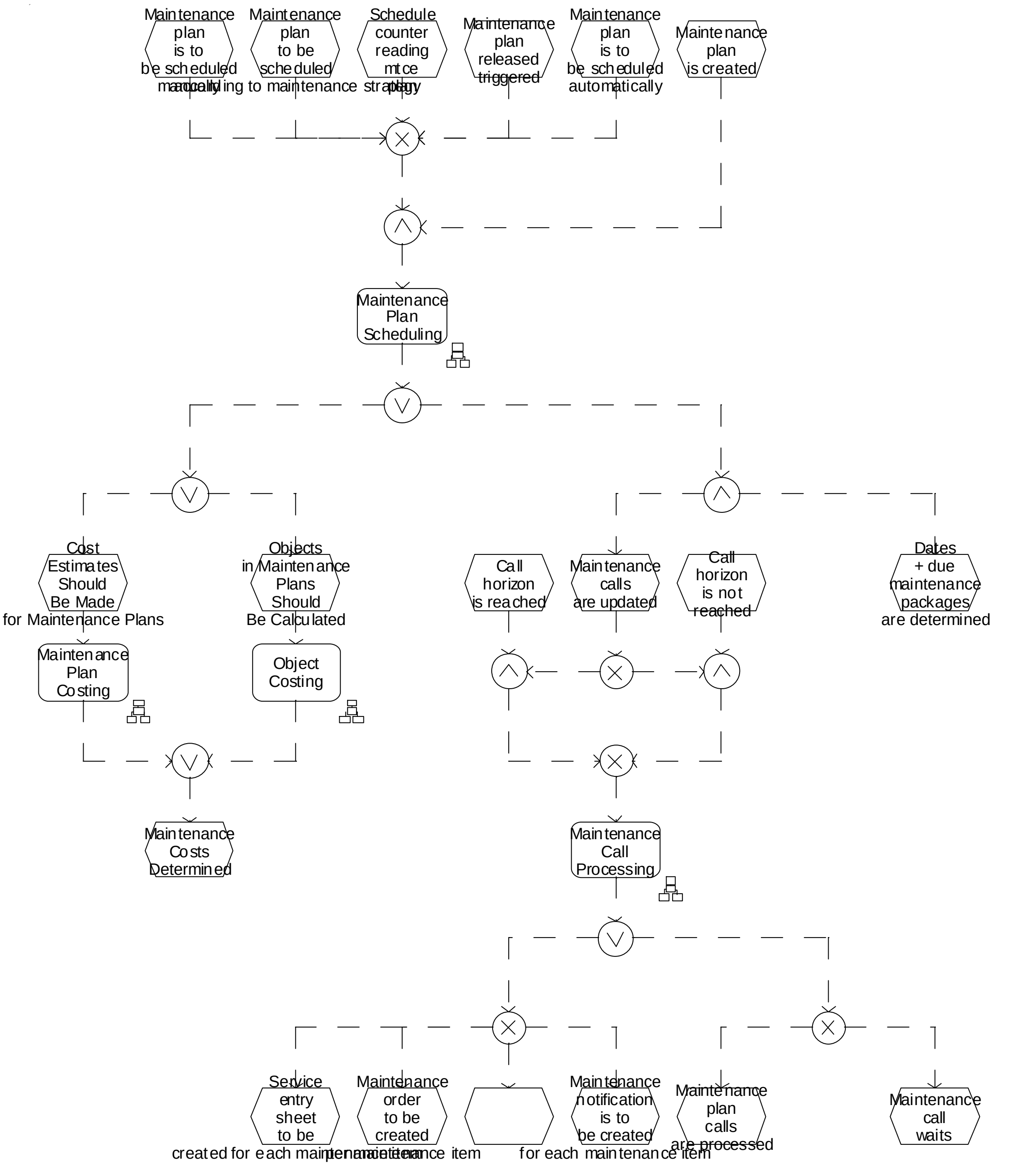


Figure B.24: Plant Maintenance – Planned Maintenance Processing – Maintenance Planning (reduced size 10, unsound)

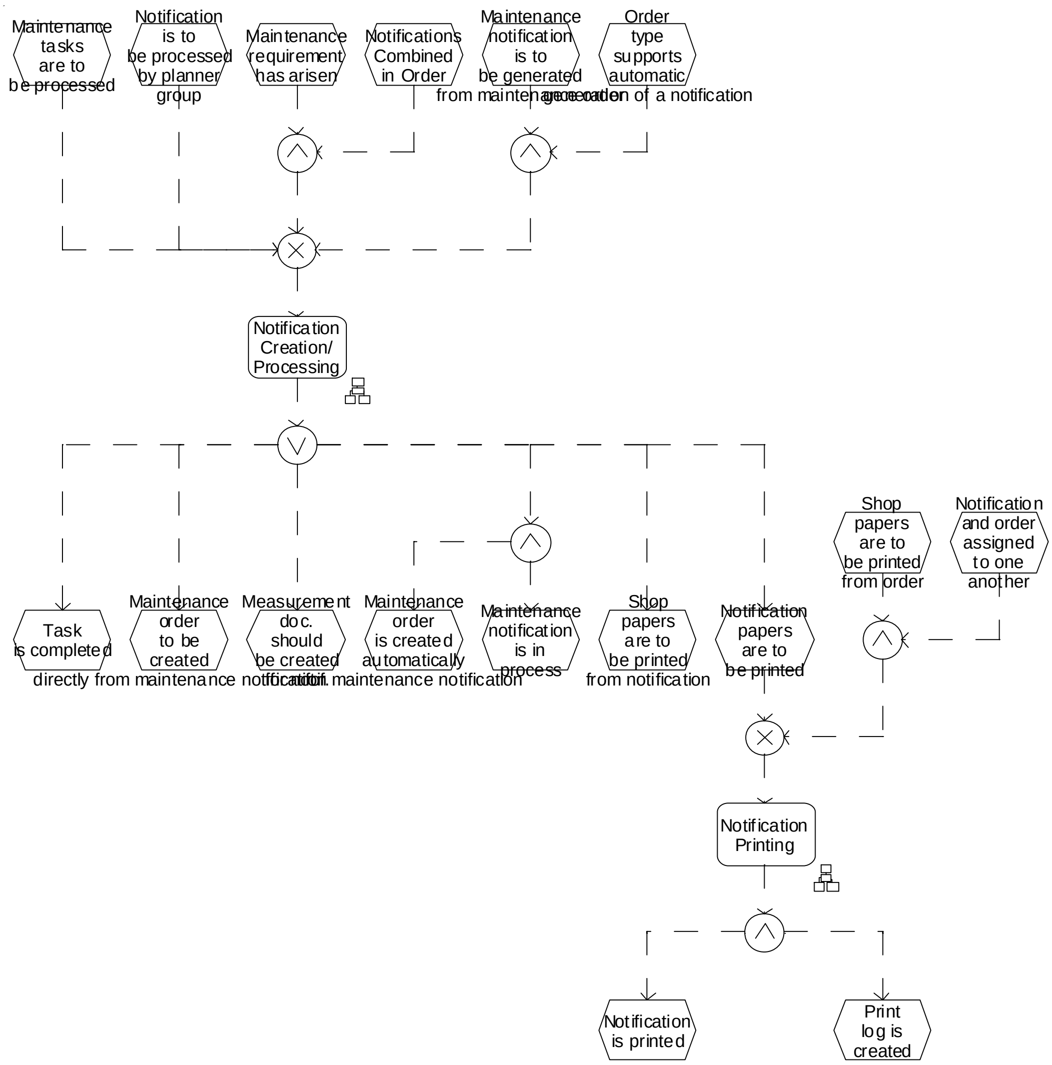


Figure B.25: Plant Maintenance – Planned Maintenance Processing – Notification (reduced size 6, sound)

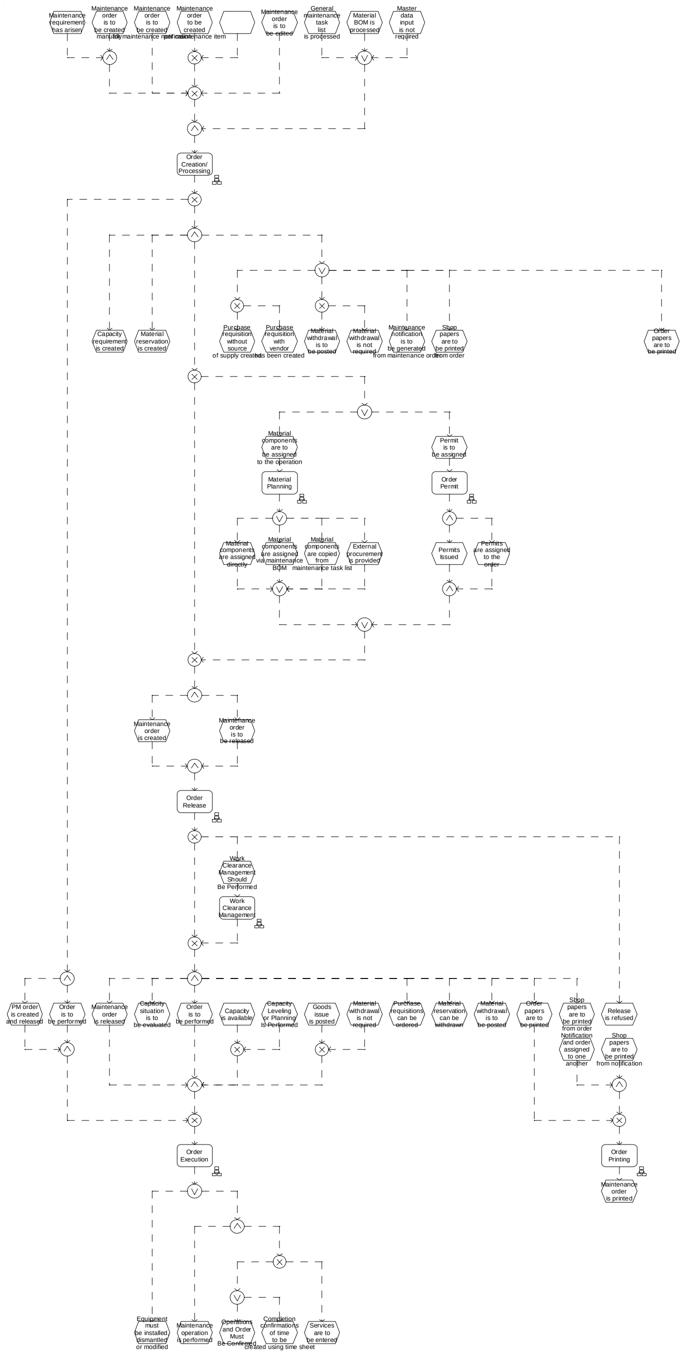


Figure B.26: Plant Maintenance – Planned Maintenance Processing – Order (reduced size 11, unsound)

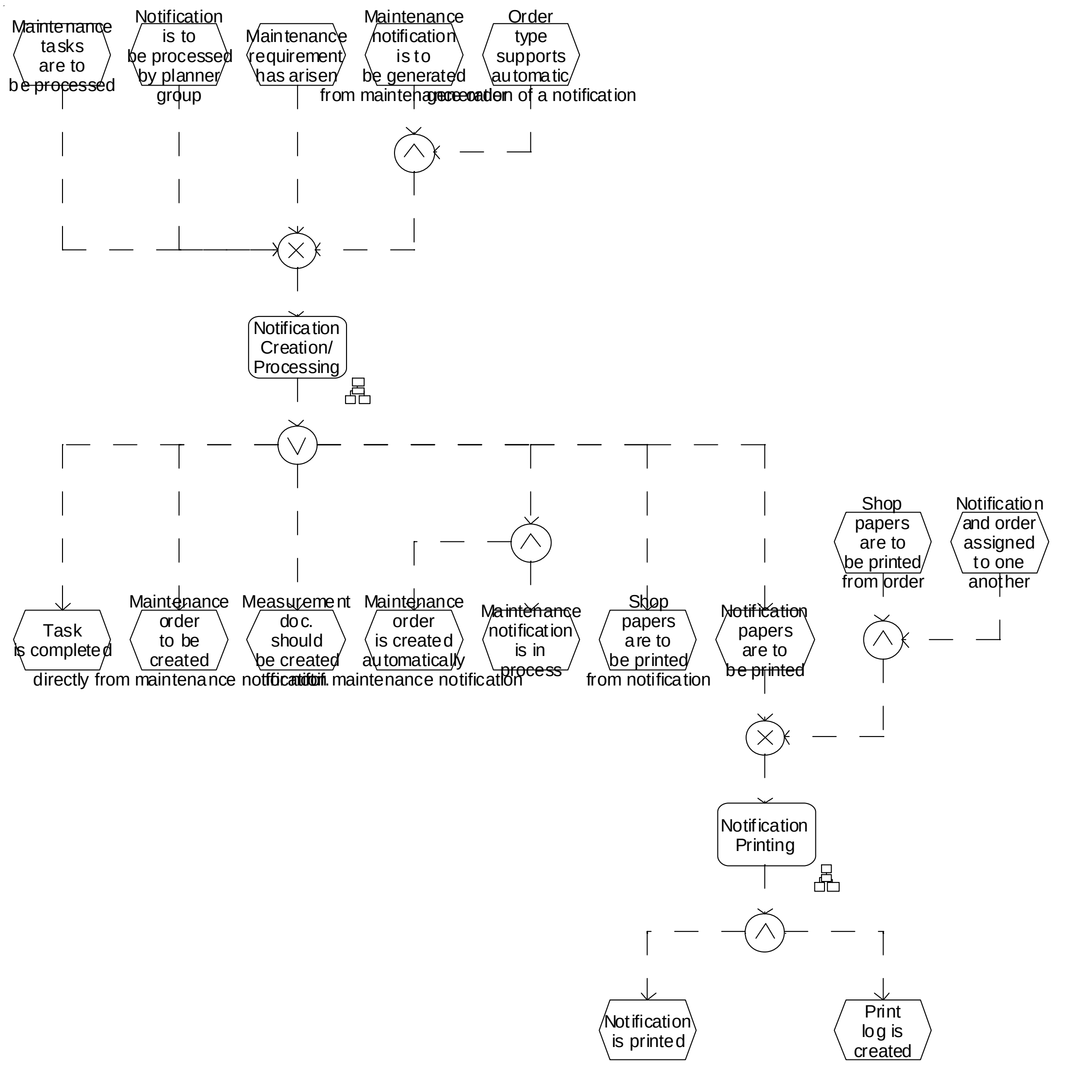


Figure B.27: Plant Maintenance – Project-Based Maintenance Processing – Notification (reduced size 6, sound)

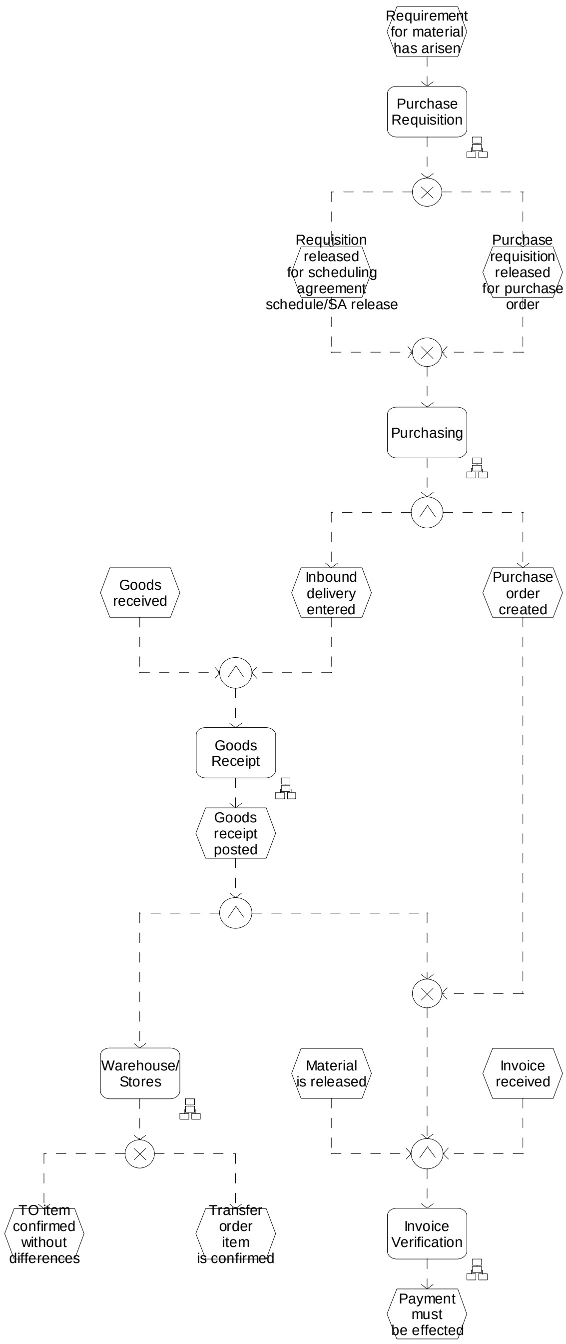


Figure B.28: Procurement – Internal Procurement (reduced size 8, unsound)

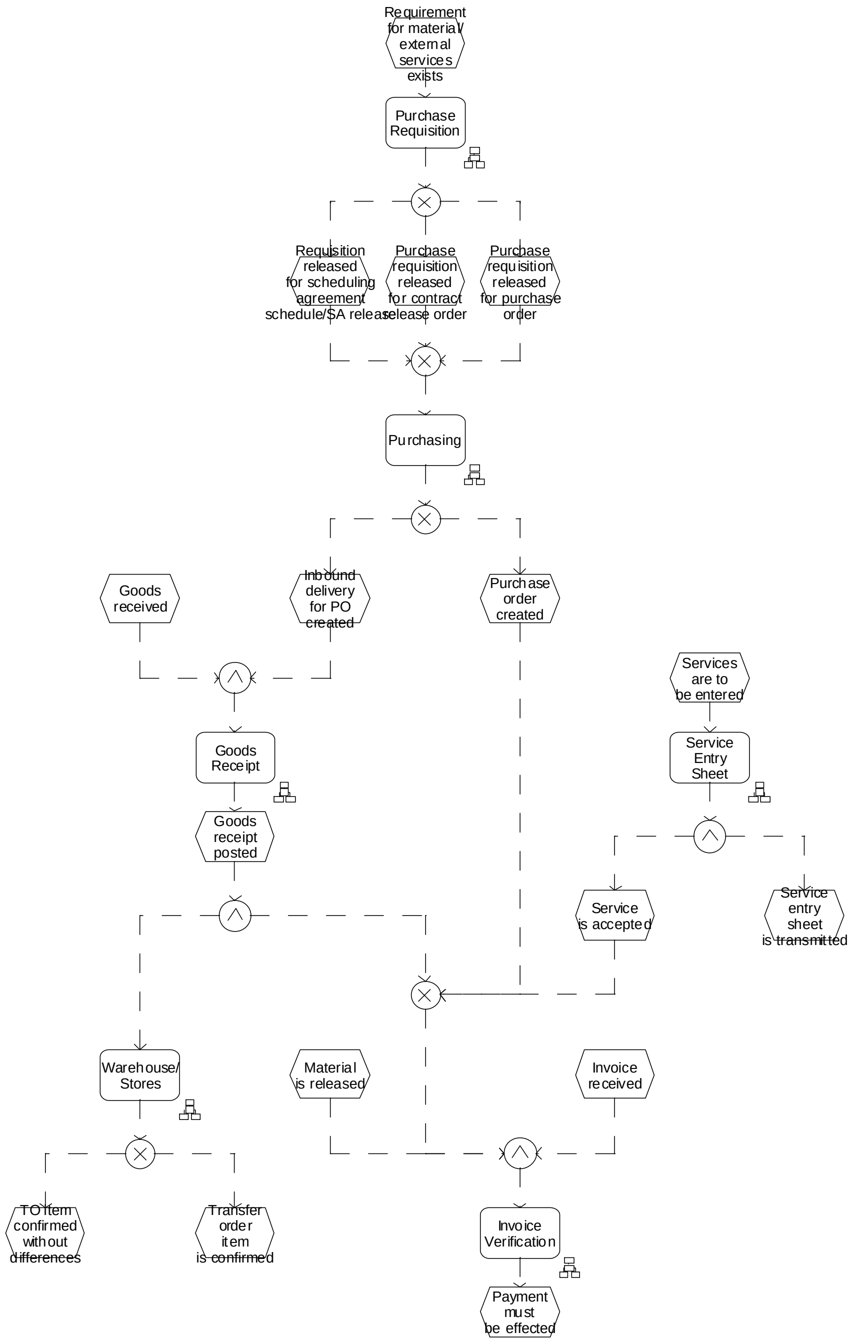


Figure B.29: Procurement – Procurement of Materials and External Services (reduced size 9, unsound)

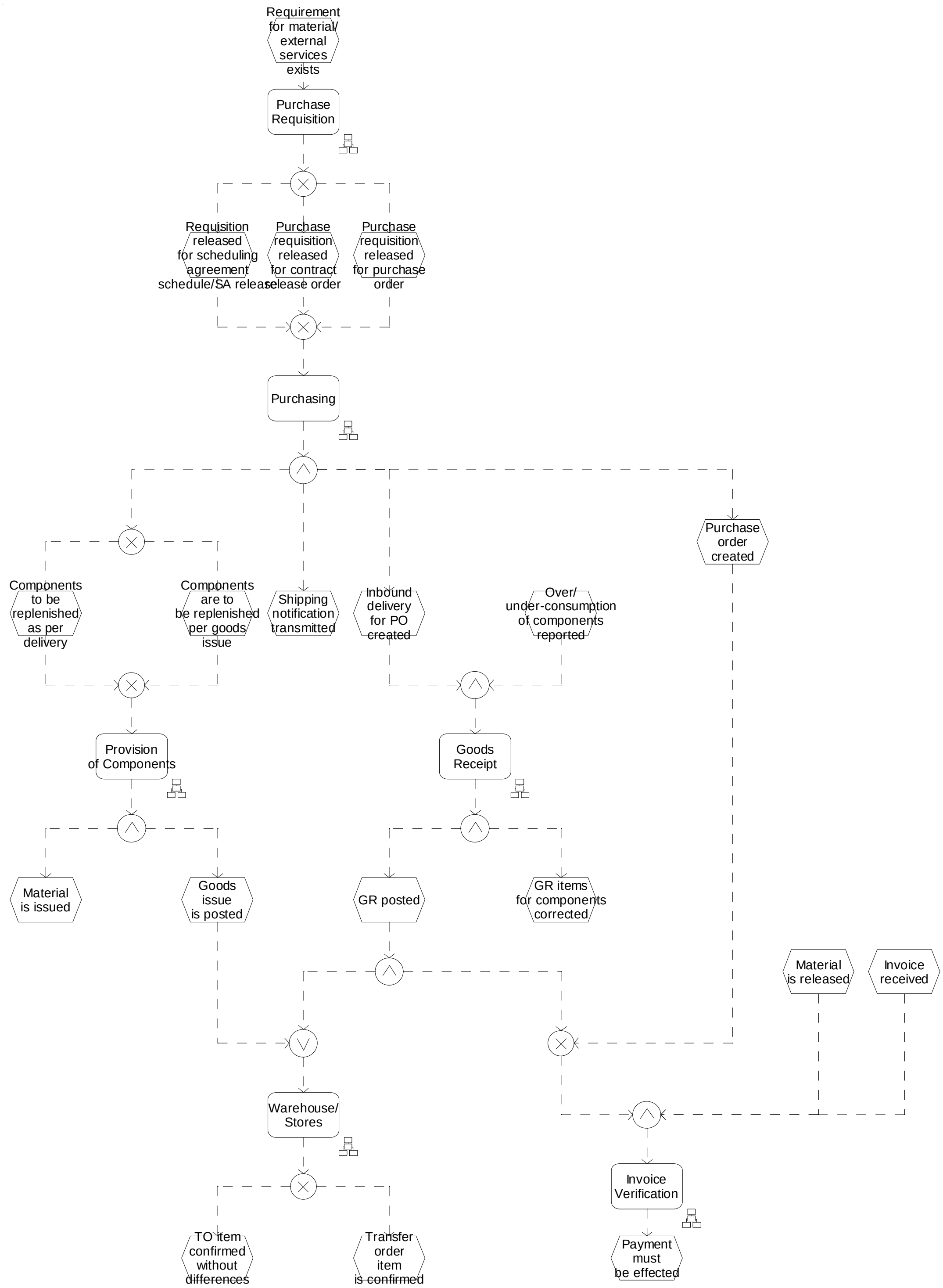


Figure B.30: Procurement – Procurement via Subcontracting (reduced size 11, unsound)

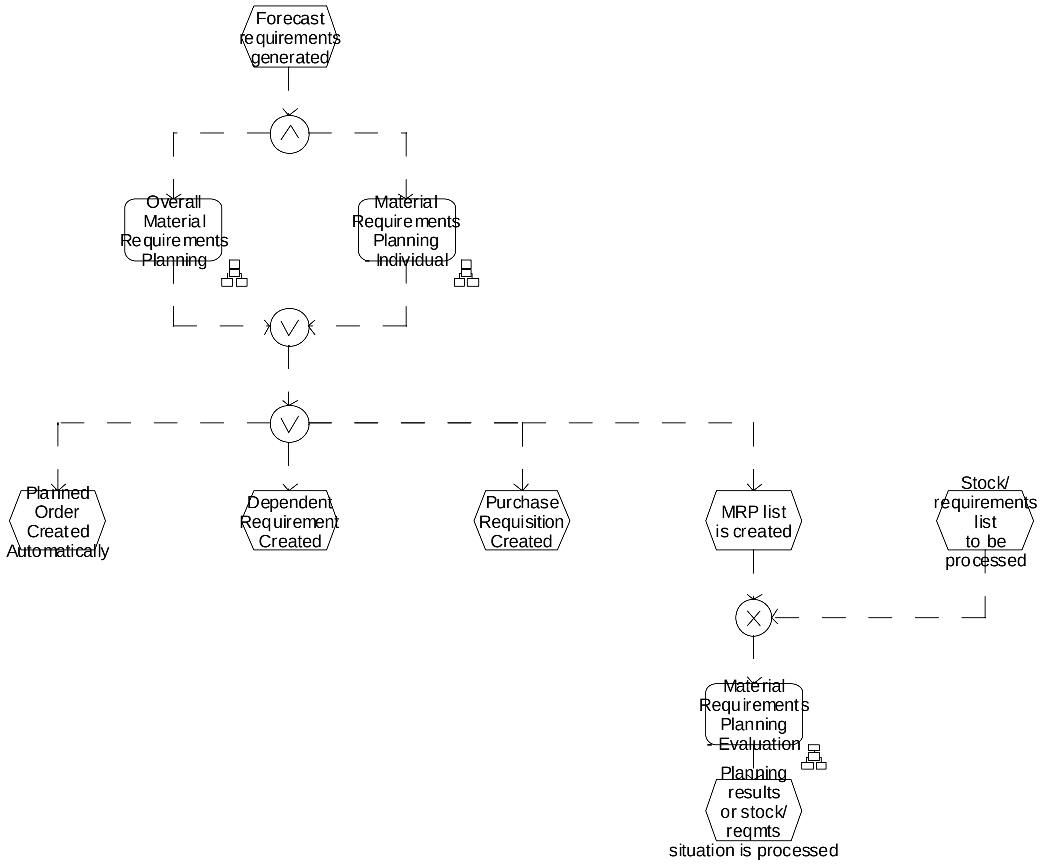


Figure B.31: Production Planning and Procurement Planning – Consumption-Driven Planning – Material Requirements Planning (reduced size 6, sound)

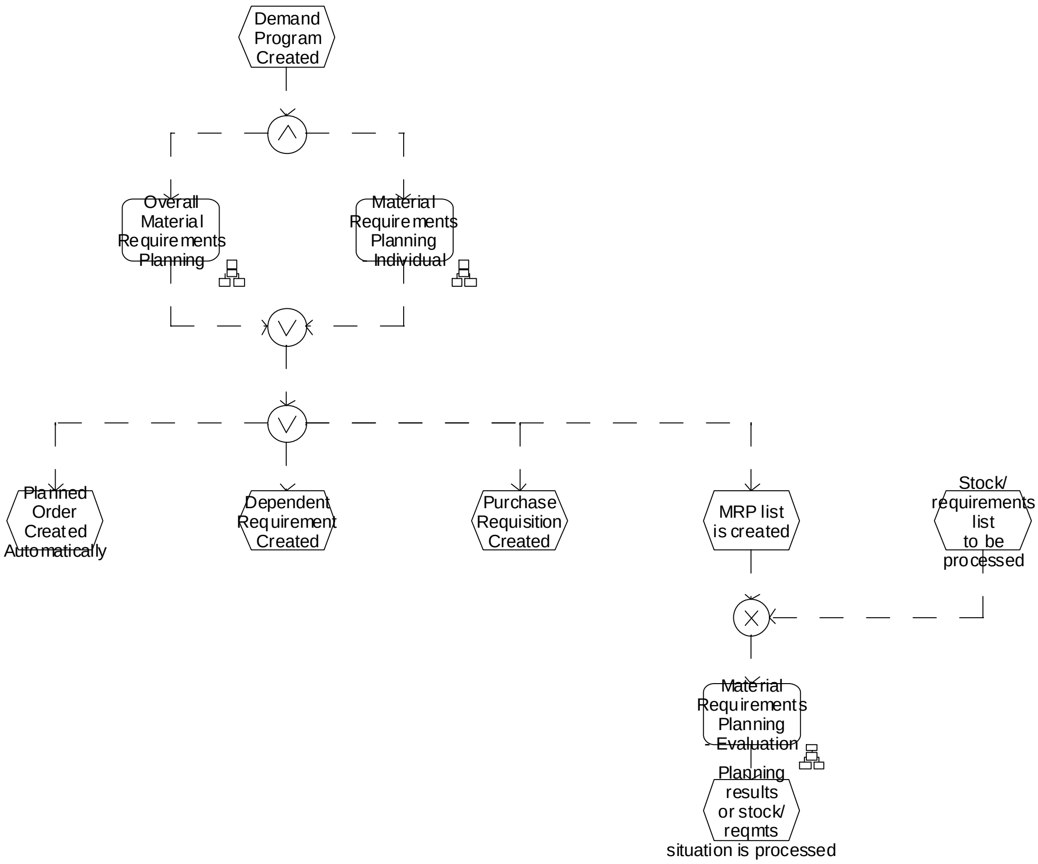


Figure B.32: Production Planning and Procurement Planning – Market-Oriented Planning (reduced size 11, sound)

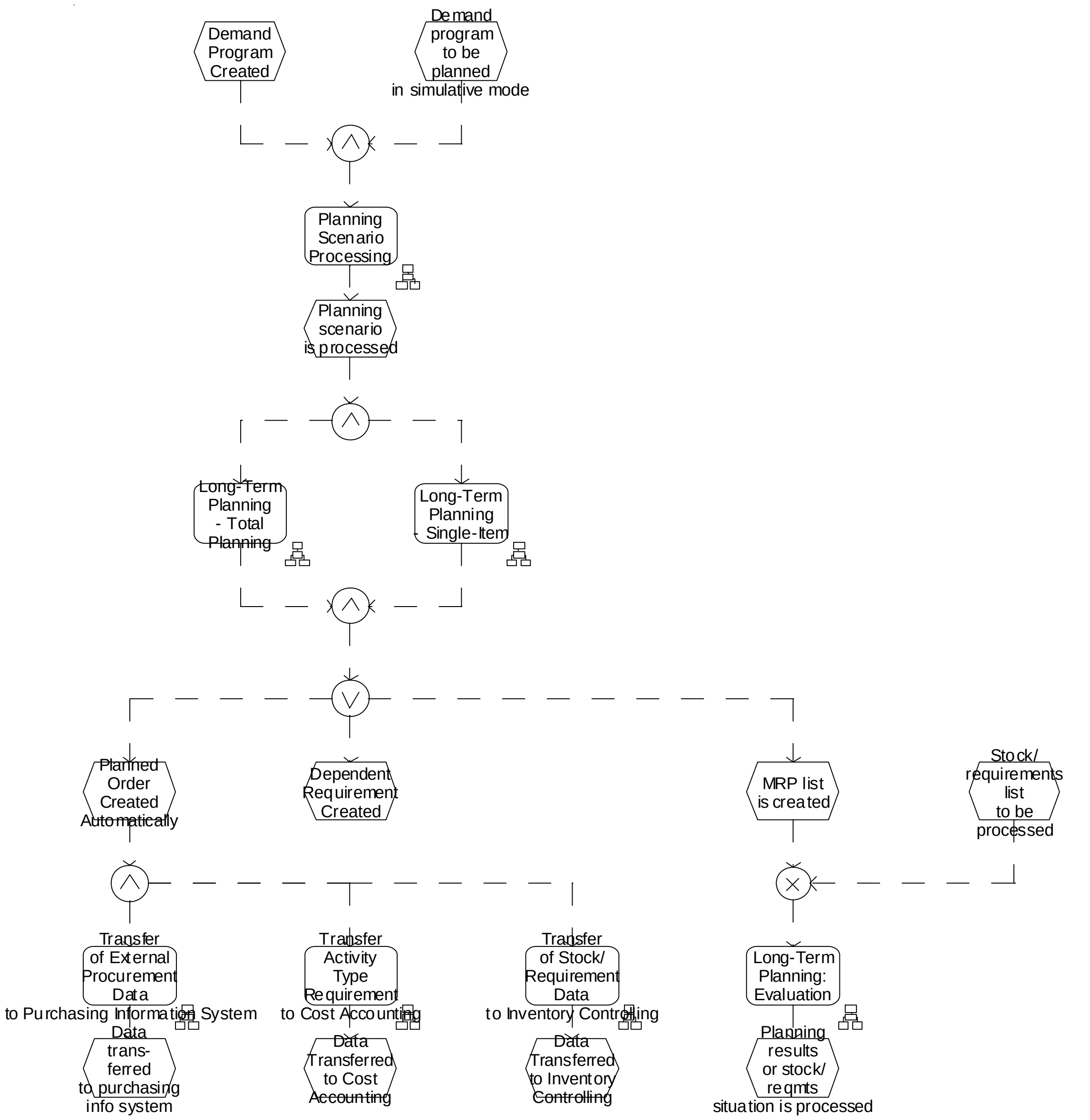


Figure B.33: Production Planning and Procurement Planning – Market-Oriented Planning – Long-Term Planning (reduced size 6, sound)

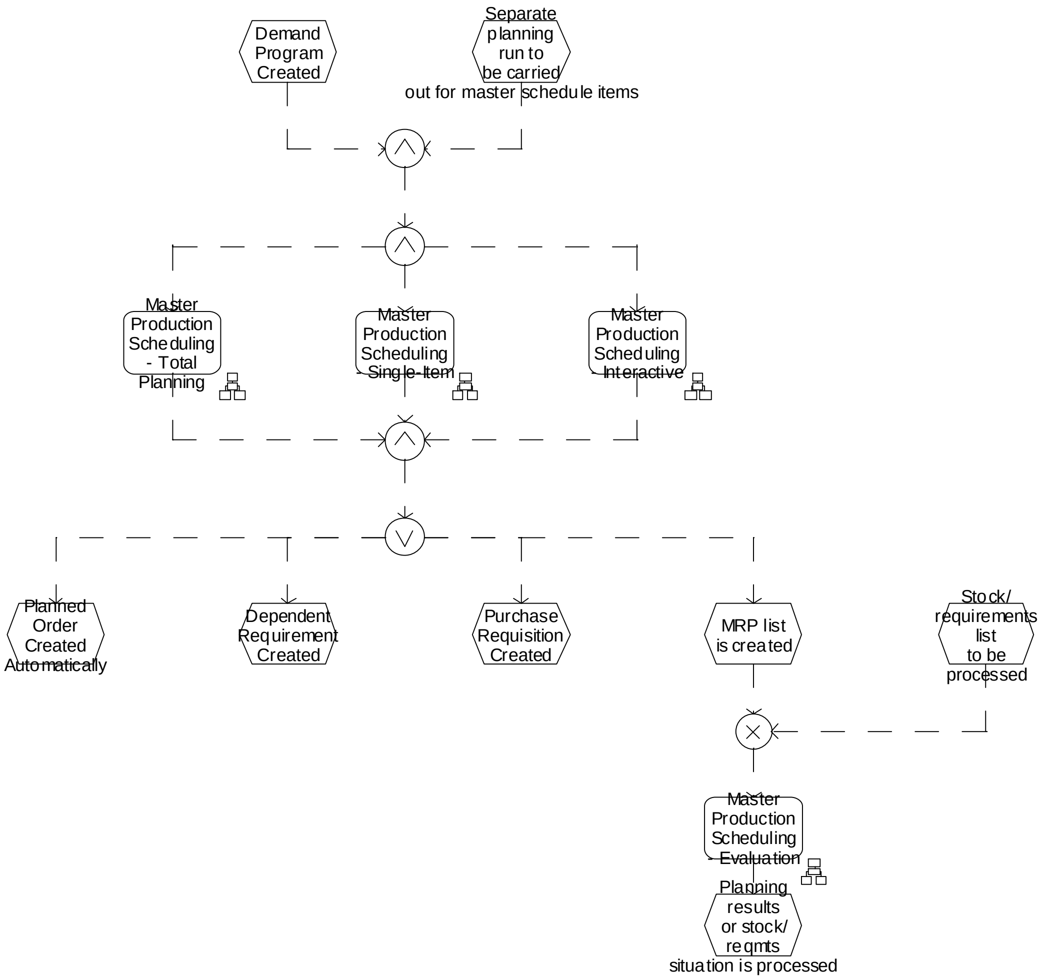


Figure B.34: Production Planning and Procurement Planning – Market-Oriented Planning – Master Production Scheduling (reduced size 6, sound)

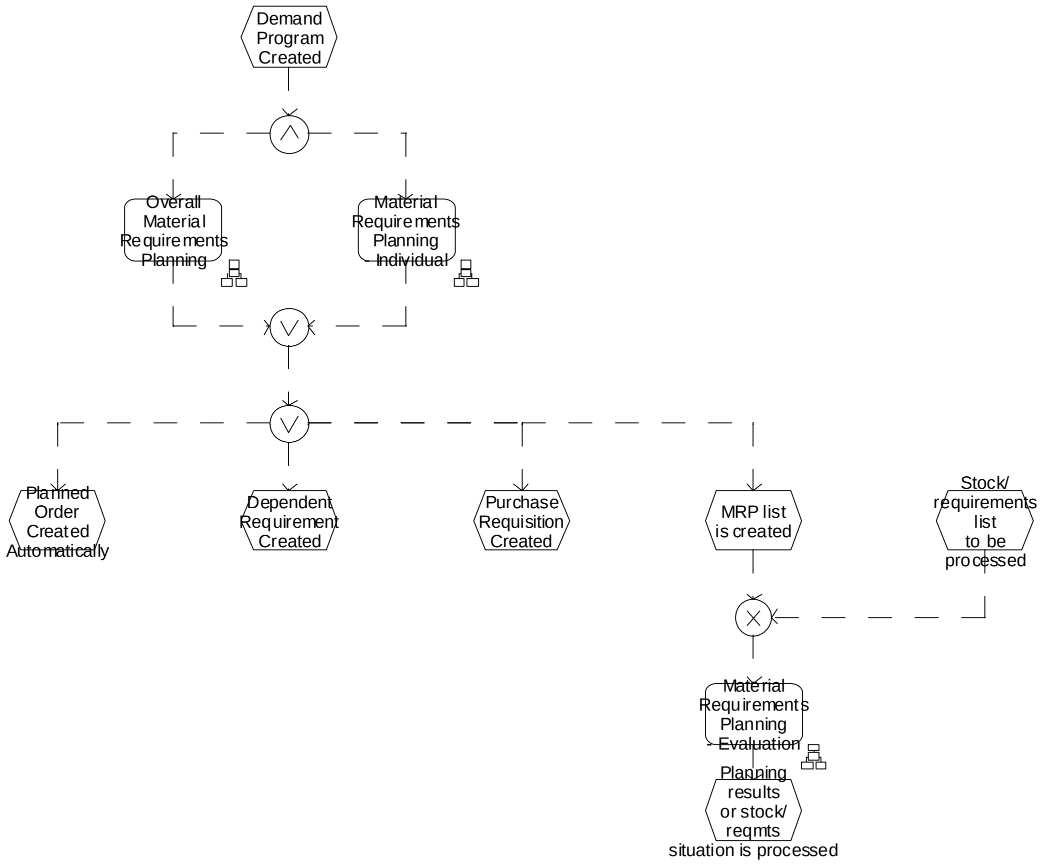


Figure B.35: Production Planning and Procurement Planning – Market-Oriented Planning – Material Requirements Planning (reduced size 6, sound)

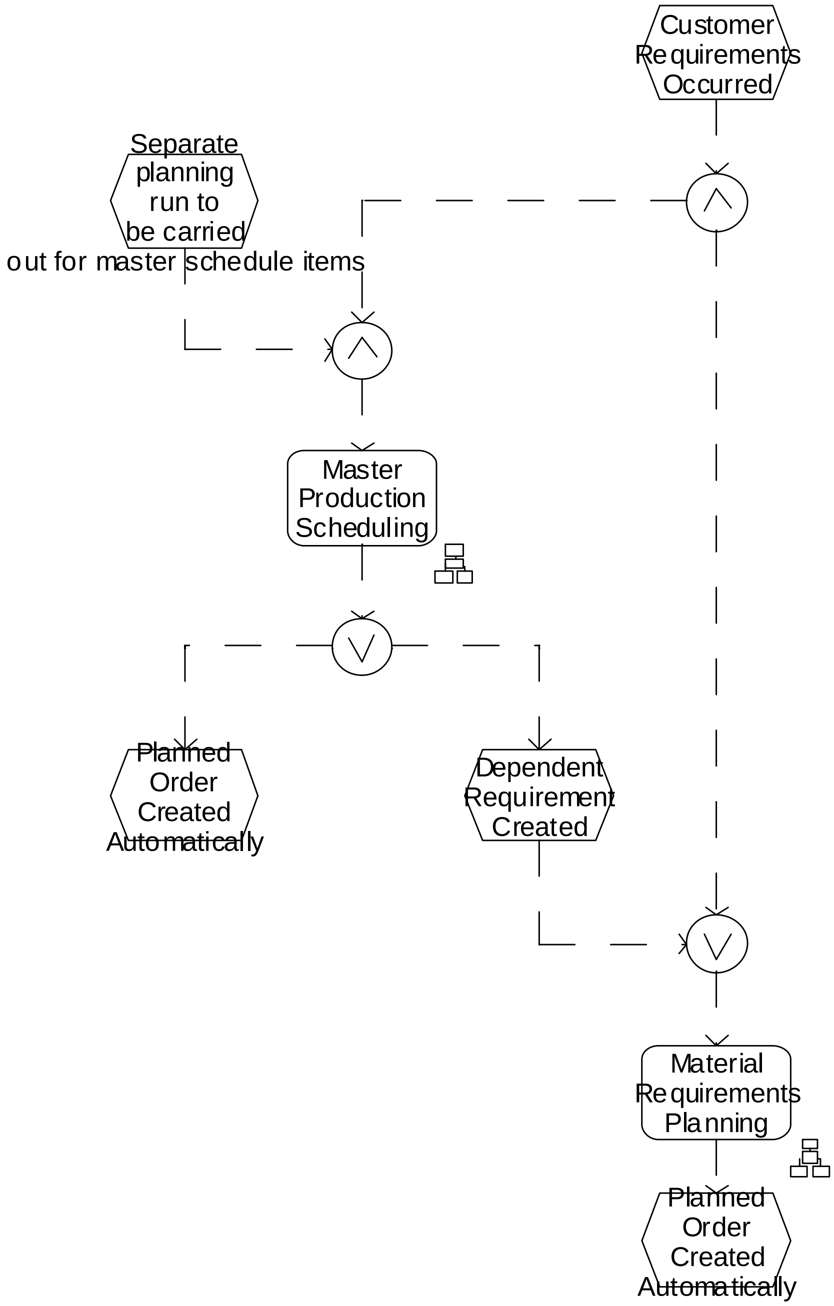


Figure B.36: Production Planning and Procurement Planning – Sales Order Oriented Planning (reduced size 8, sound)

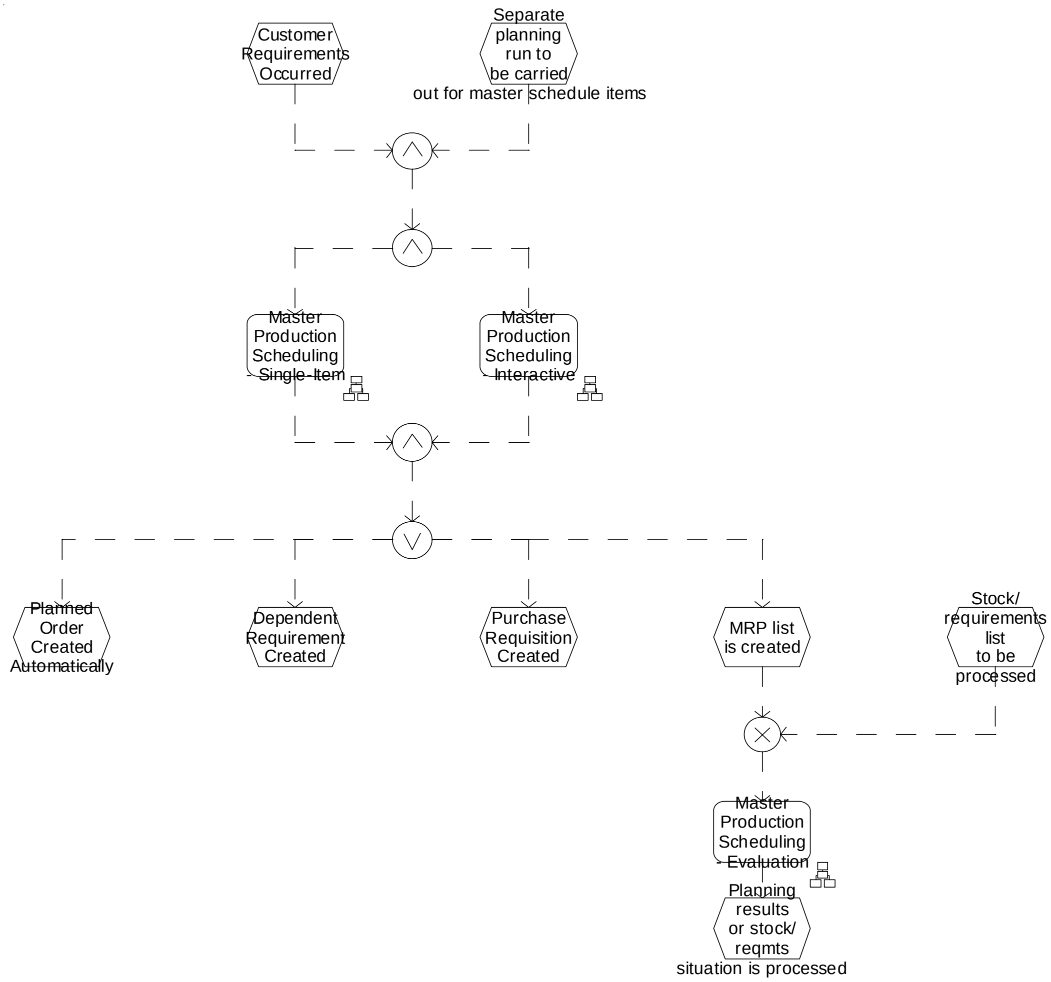


Figure B.37: Production Planning and Procurement Planning – Sales Order Oriented Planning – Master Production Scheduling (reduced size 6, sound)

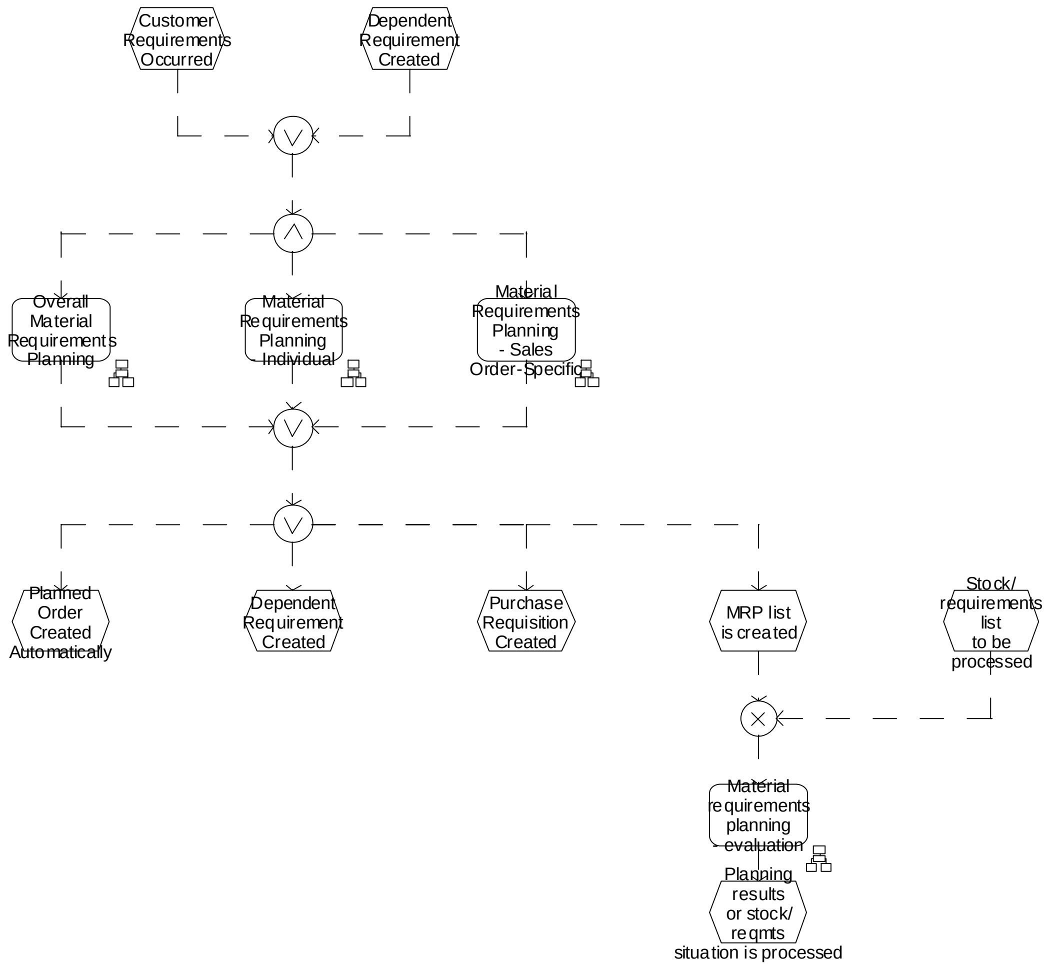


Figure B.38: Production Planning and Procurement Planning – Sales Order Oriented Planning – Material Requirements Planning (reduced size 6, sound)

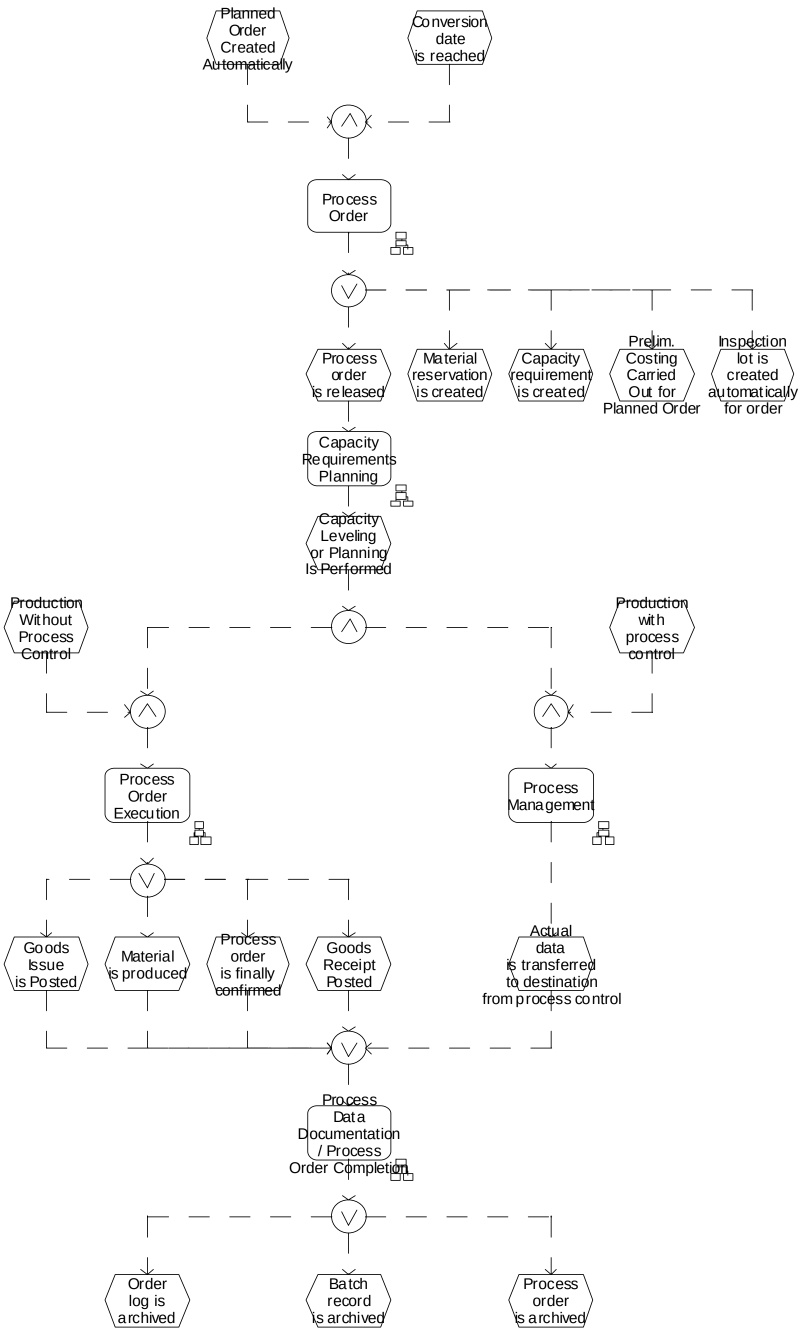


Figure B.39: Production – Process Manufacturing (reduced size 10, unsound)

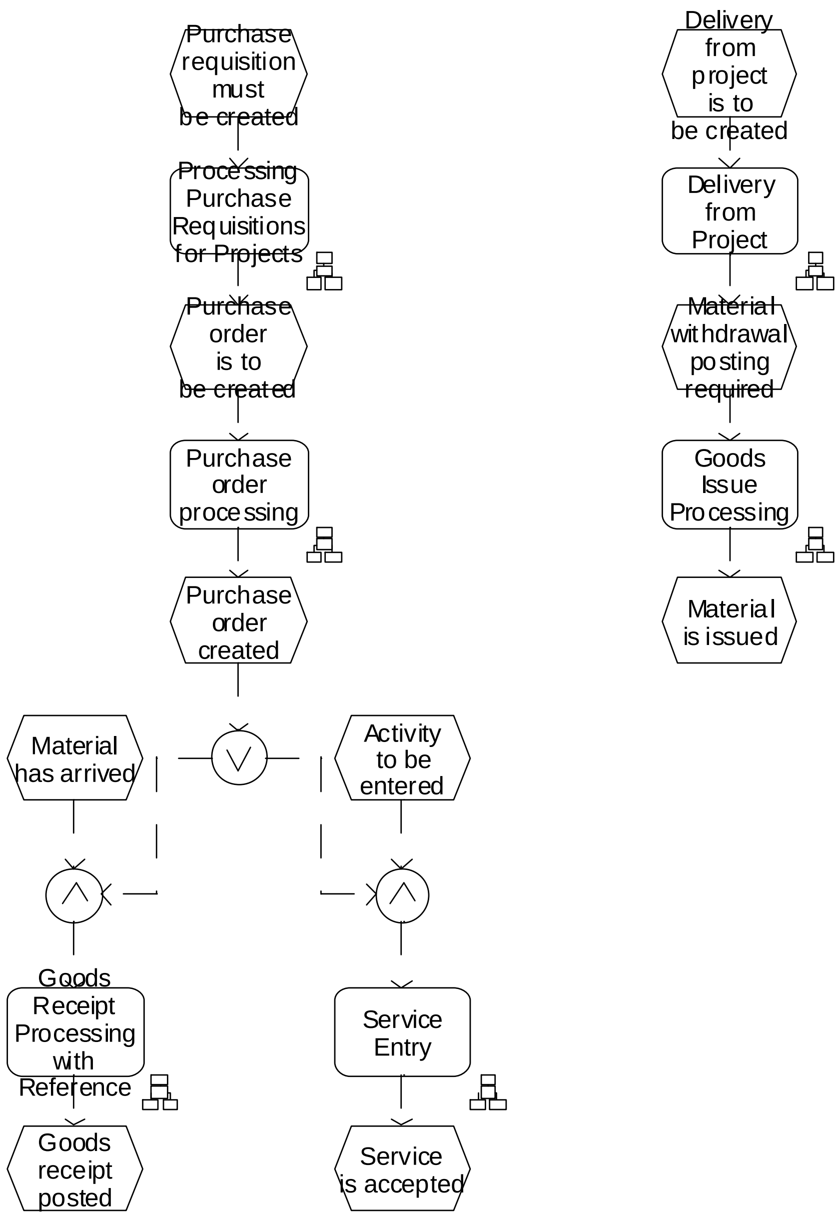


Figure B.40: Project Management – Execution – Materials Procurement and Service Processing (reduced size 8, unsound)

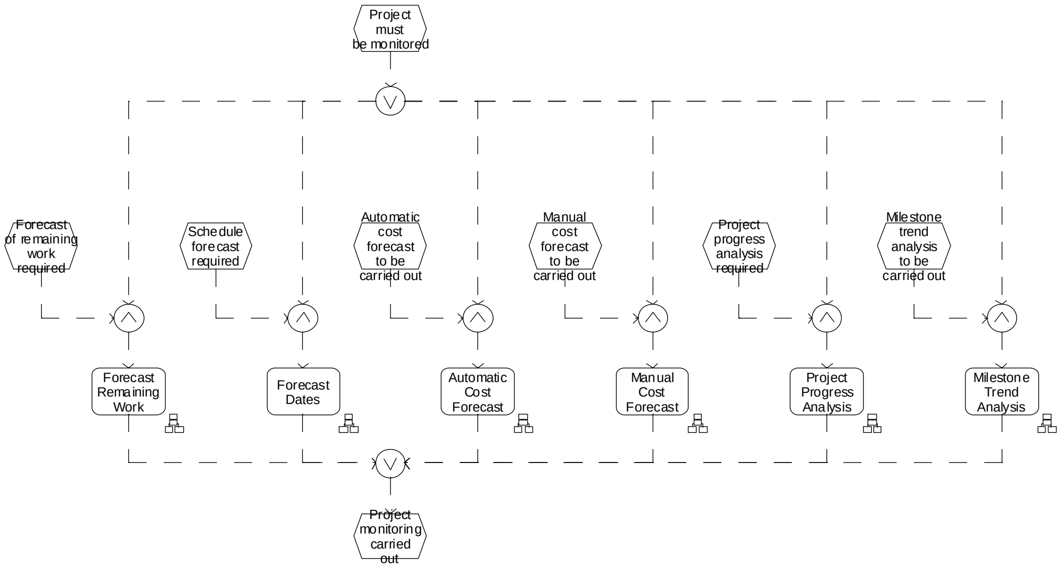


Figure B.41: Project Management – Execution – Project Monitoring and Controlling (reduced size 16, unsound)

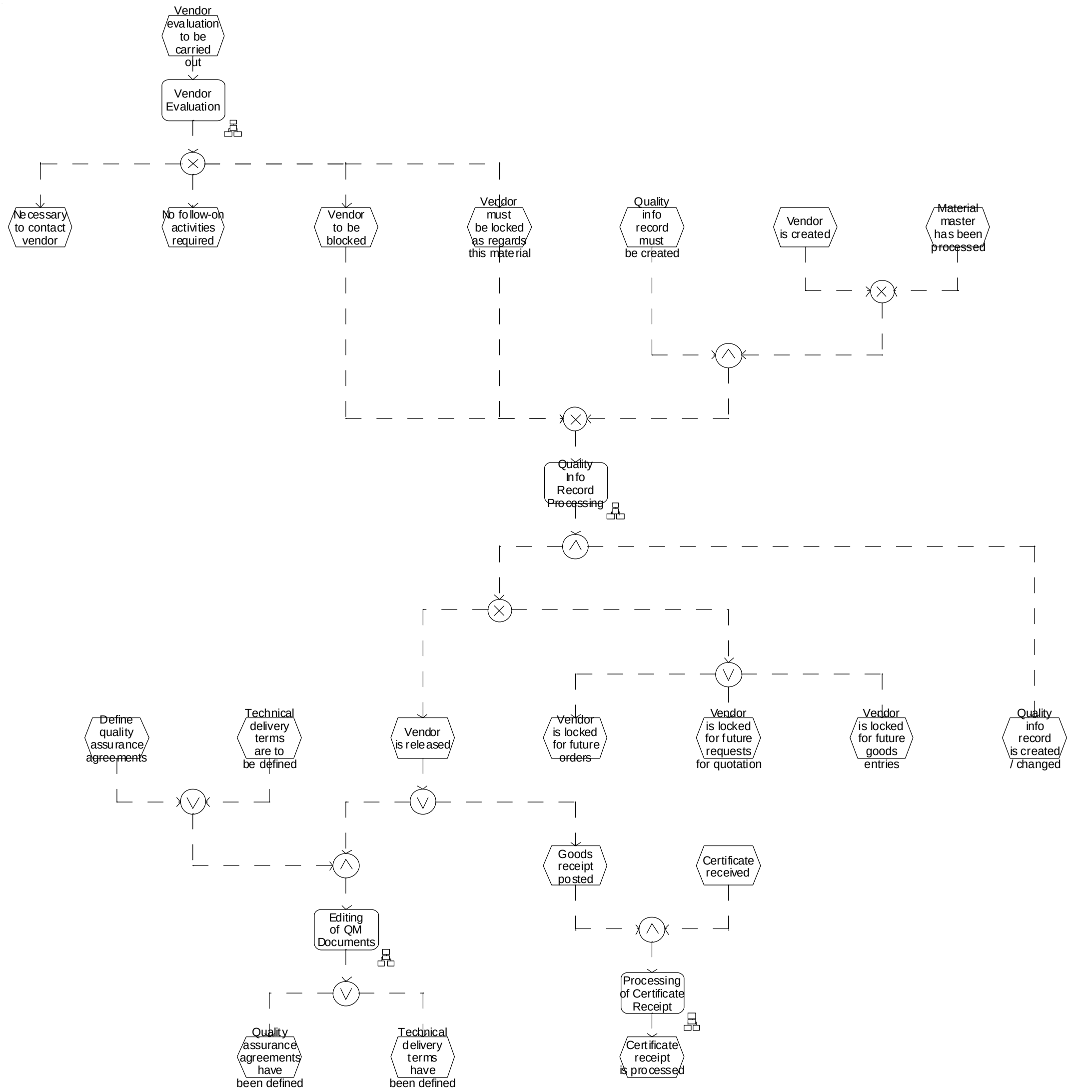


Figure B.42: Quality Management – QM in Materials Management – Procurement and Purchasing (reduced size 14, unsound)

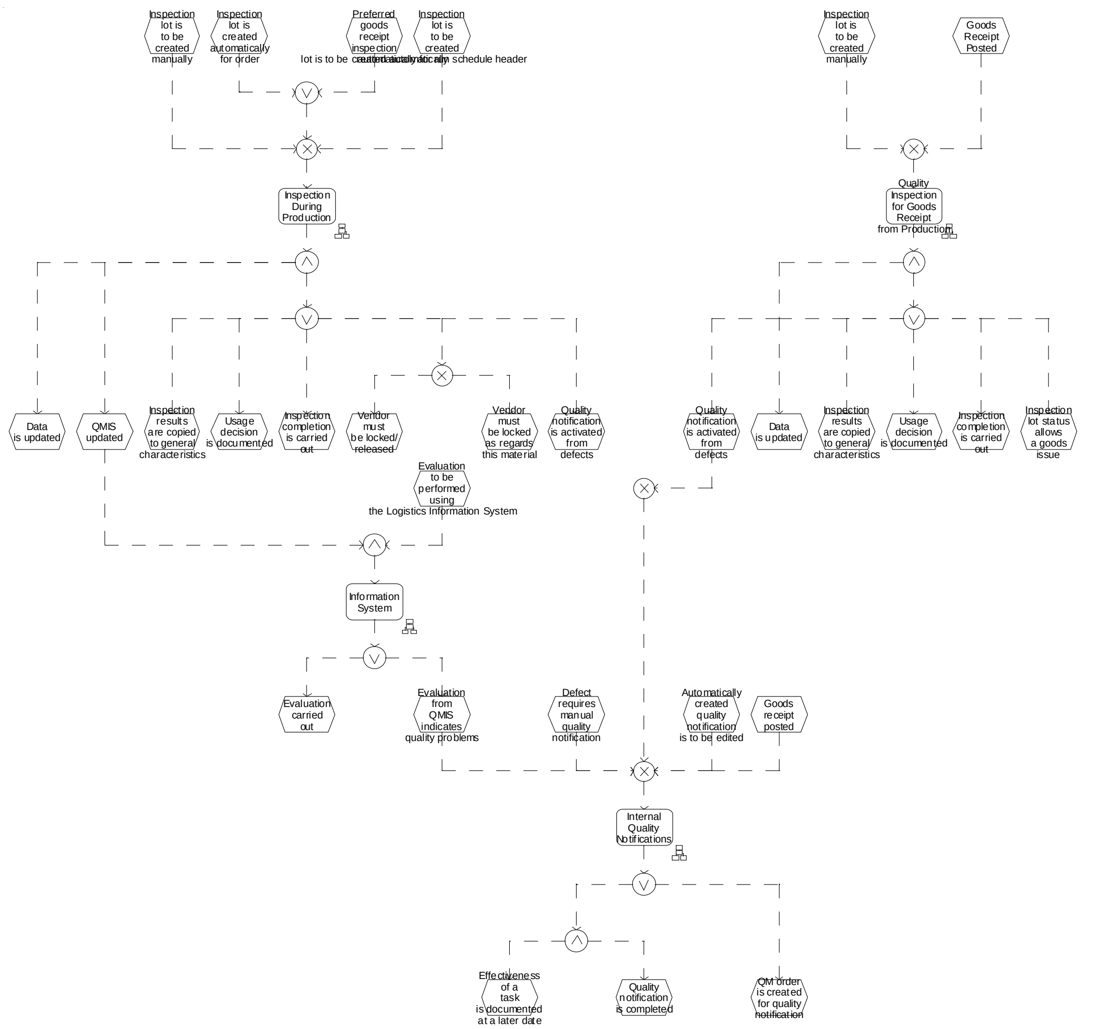


Figure B.43: Quality Management – QM in Production (reduced size 9, sound)

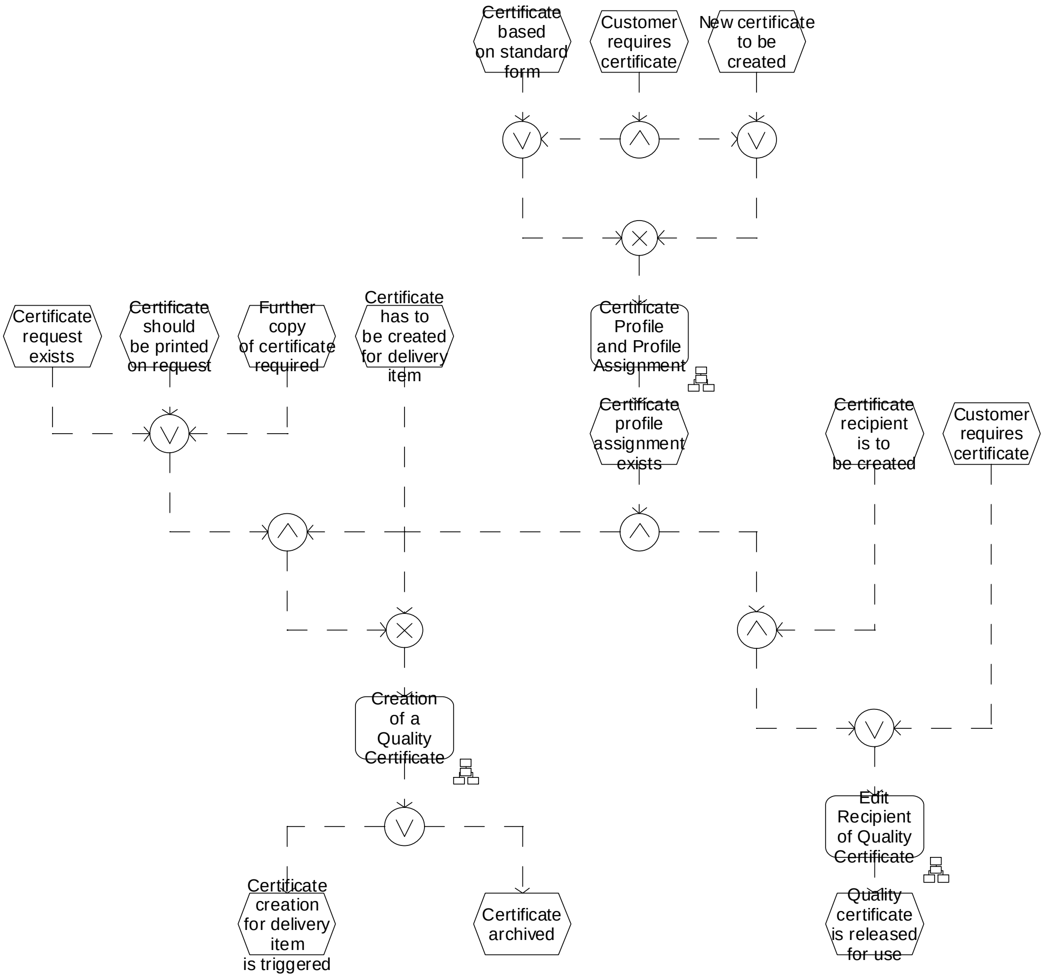


Figure B.44: Quality Management – QM in Sales and Distribution – Certificate Creation (reduced size 16, unsound)

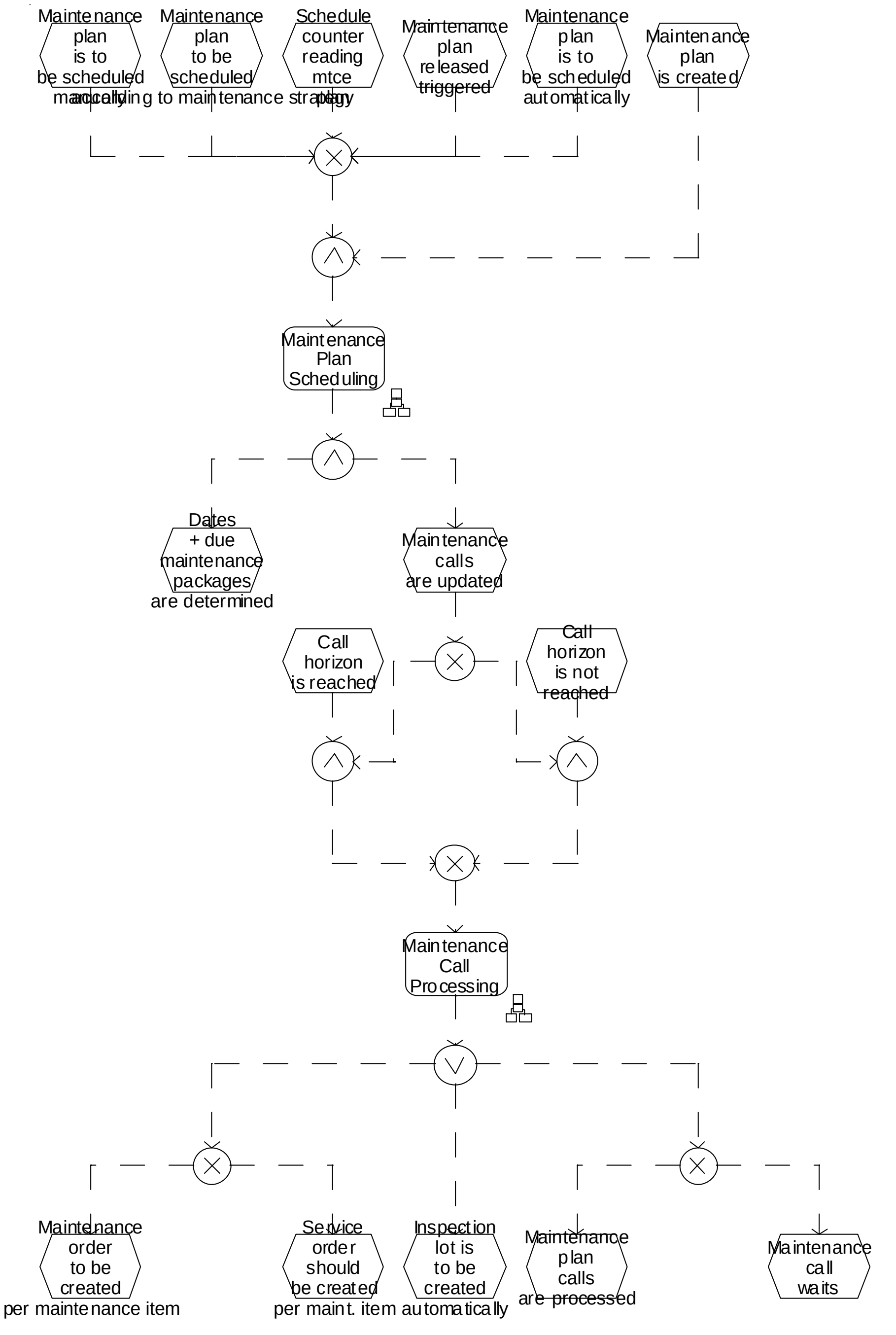


Figure B.45: Quality Management – Test Equipment Management – Maintenance Planning (reduced size 8, unsound)

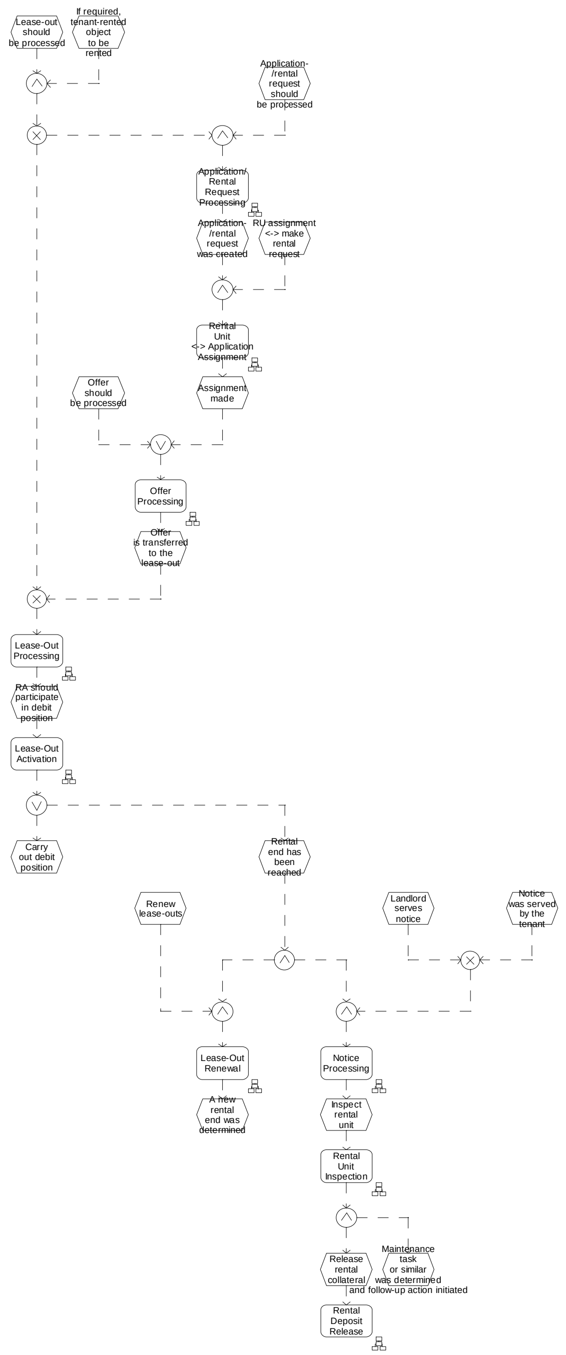


Figure B.46: Real Estate Management – Real Estate Management – Rental (reduced size 16, unsound)

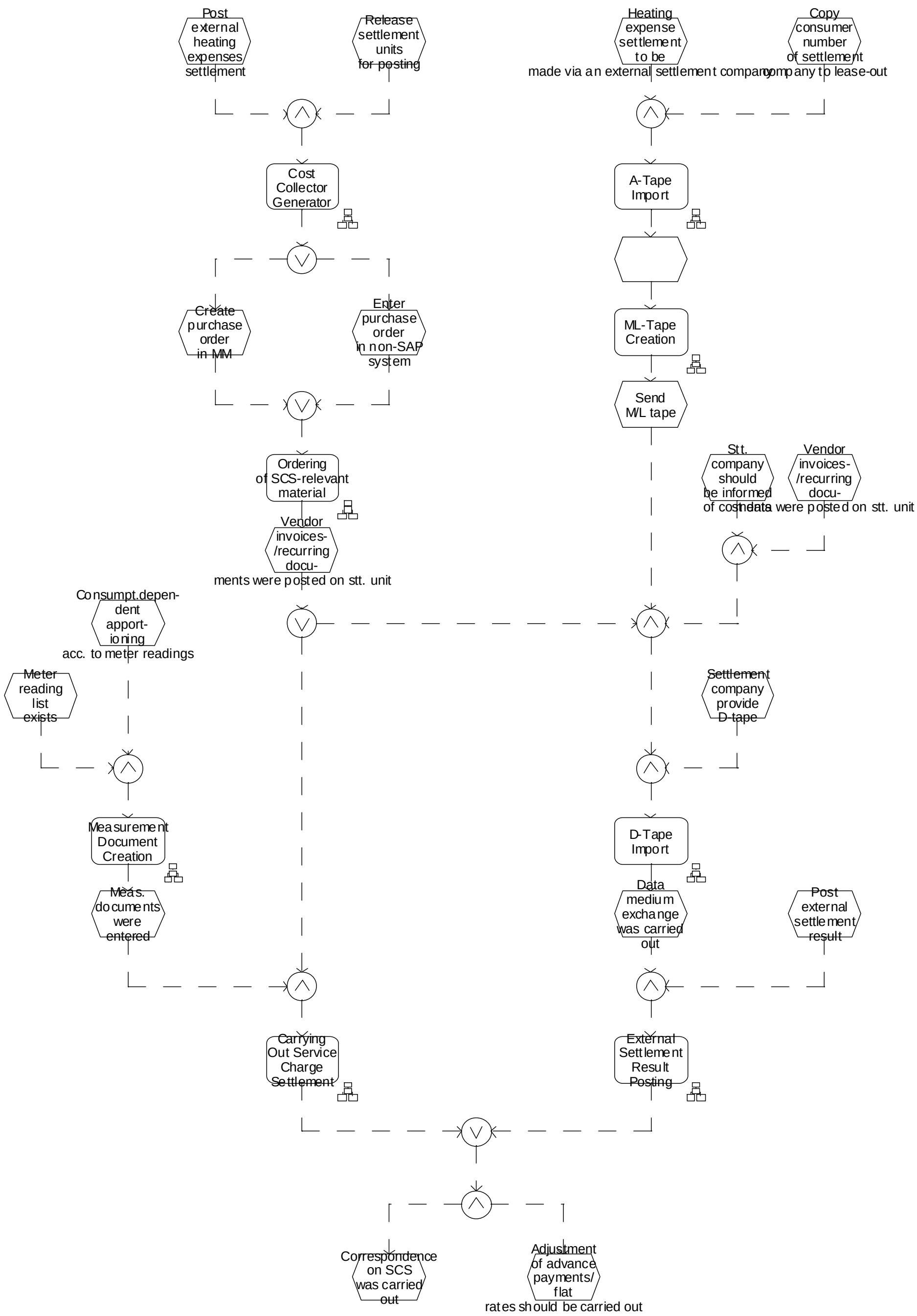


Figure B.47: Real Estate Management – Real Estate Management – Service Charge Settlement (reduced size 8, unsound)

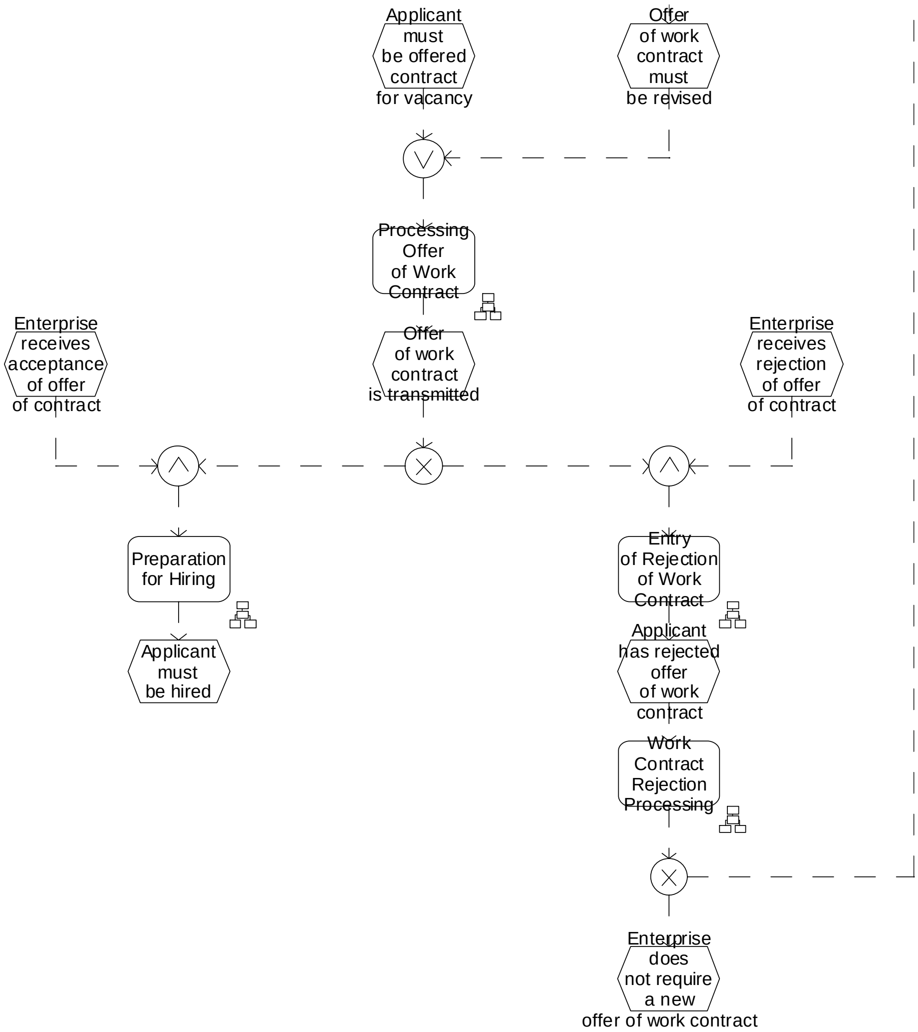


Figure B.48: Recruitment – Recruitment – Work Contract Negotiation (reduced size 10, unsound)

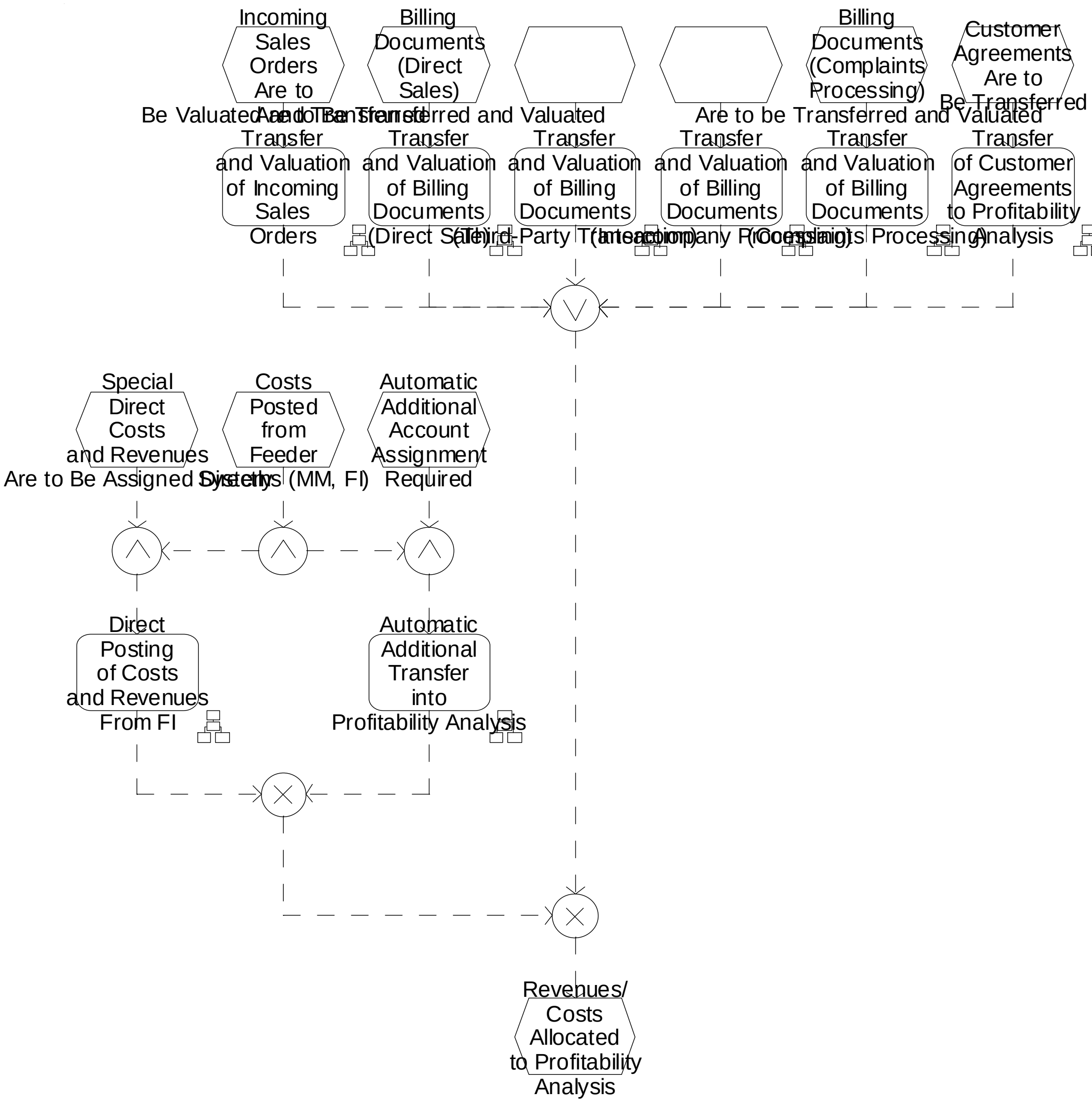


Figure B.49: Revenue and Cost Controlling – Actual Cost/Revenue Allocation – Cost and Revenue Allocation to Profitability Analysis (reduced size 9, unsound)

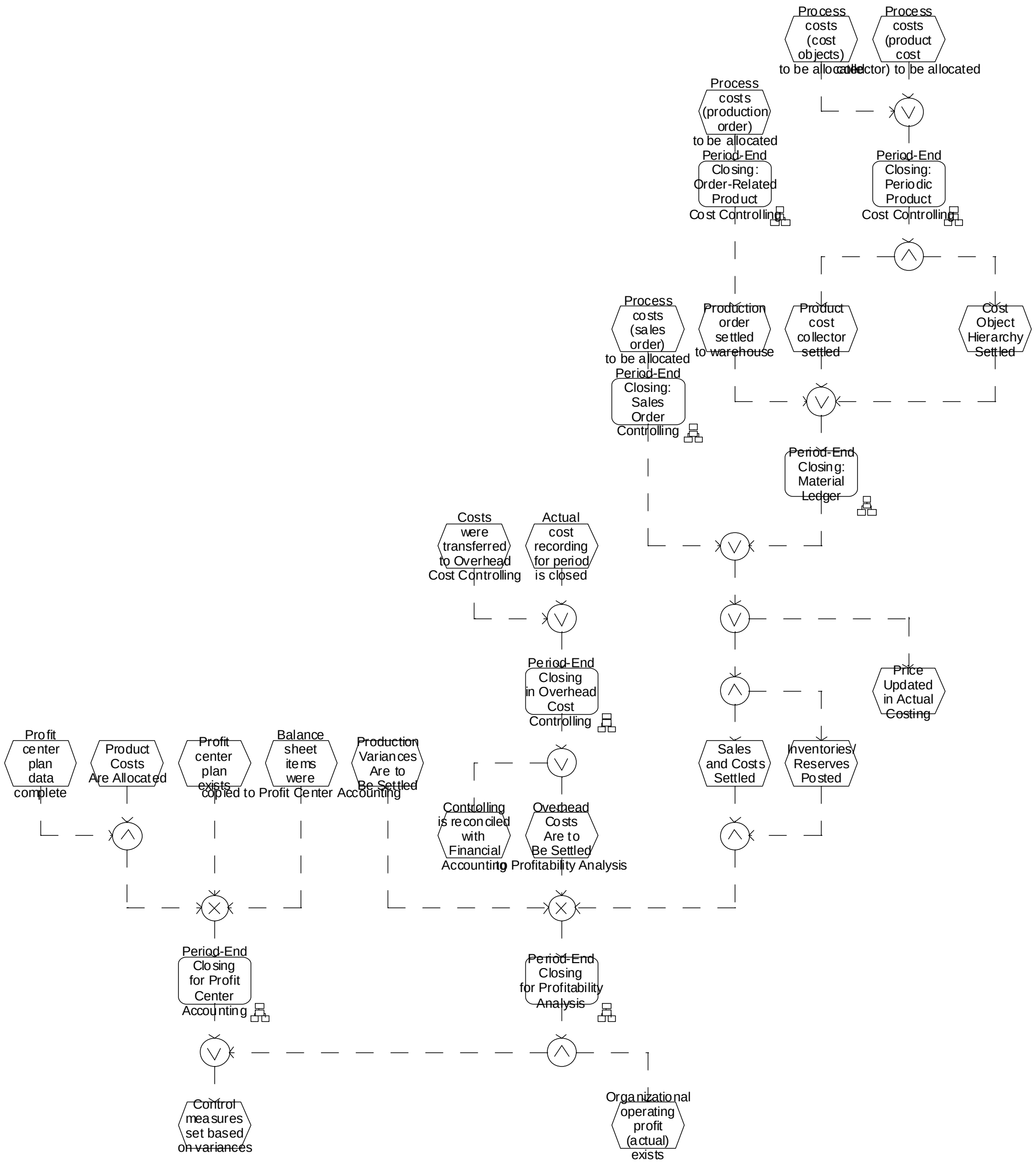


Figure B.50: Revenue and Cost Controlling – Period-End Closing (Controlling) (reduced size 11, sound)

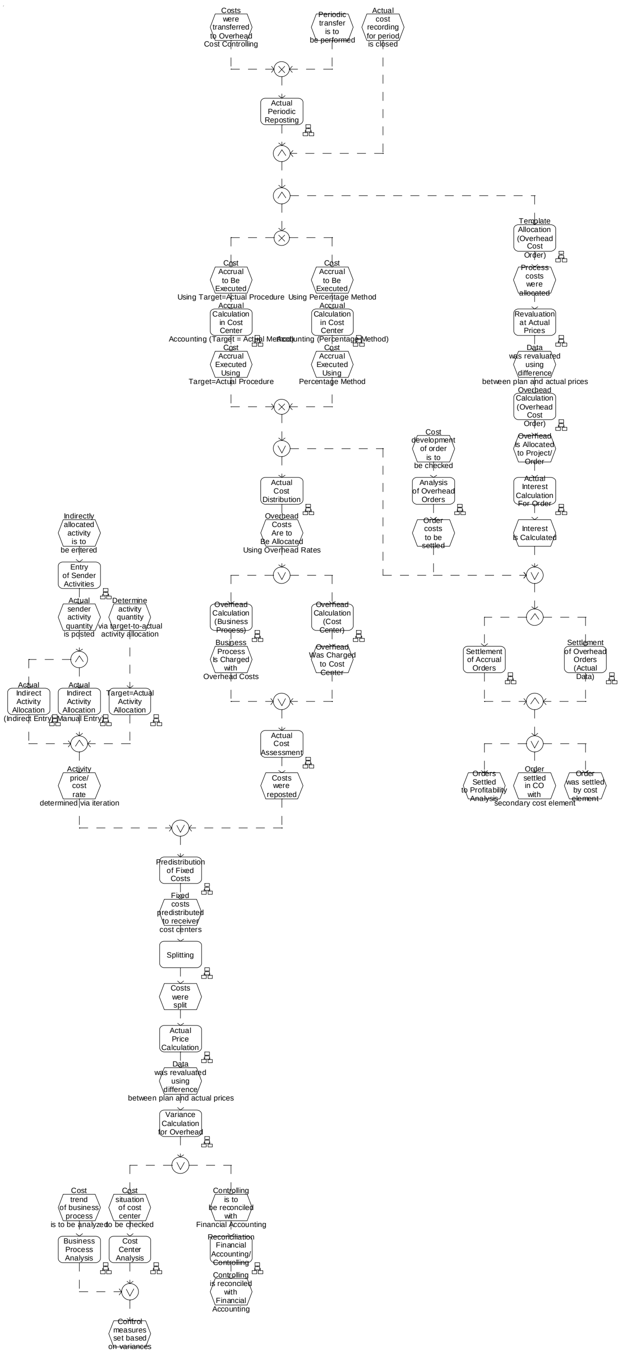


Figure B.51: Revenue and Cost Controlling – Period-End Closing (Controlling) – Period-End Closing in Overhead Cost Controlling (reduced size 13, sound)

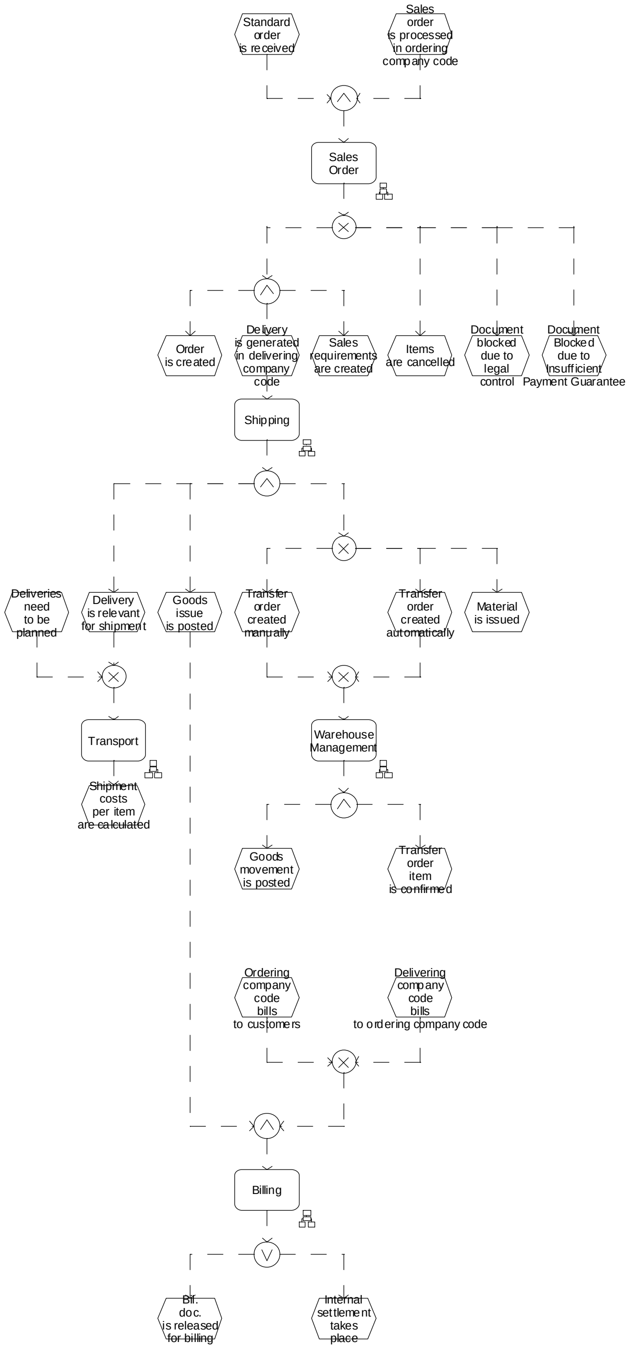


Figure B.52: Sales and Distribution – Intercompany Handling (reduced size 10, unsound)

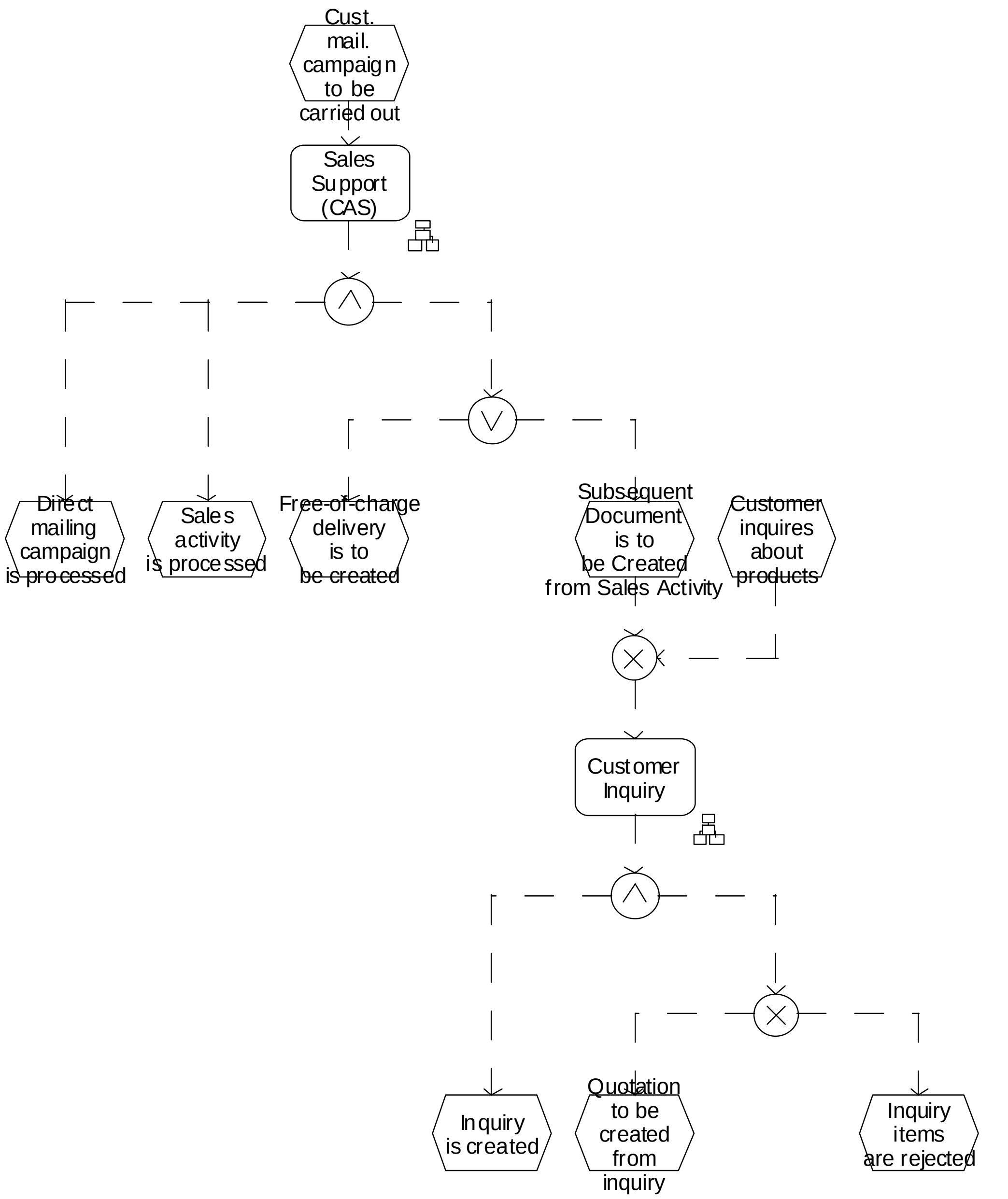


Figure B.53: Sales and Distribution – Pre-Sales Handling (reduced size 6, sound)

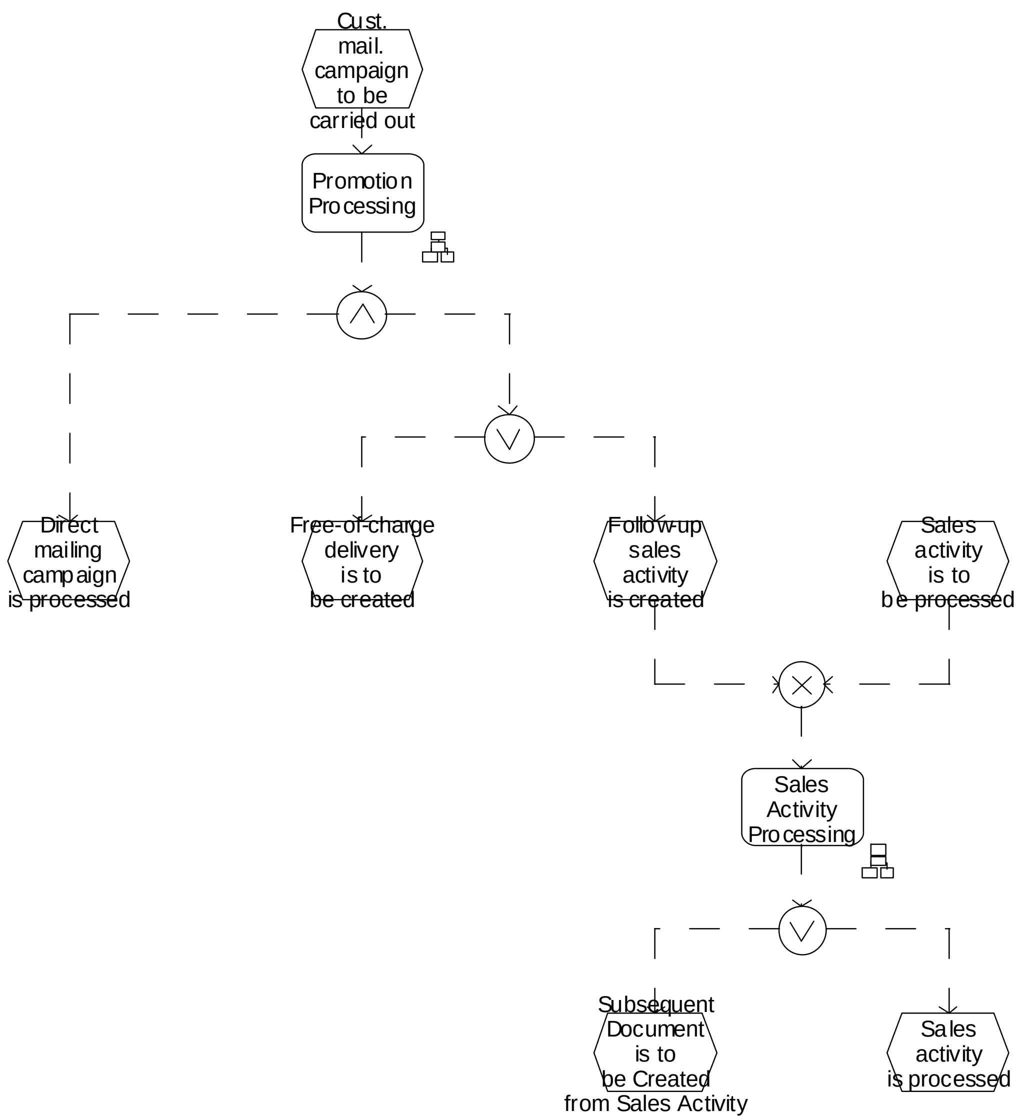


Figure B.54: Sales and Distribution – Pre-Sales Handling – Sales Support (CAS) (reduced size 6, sound)

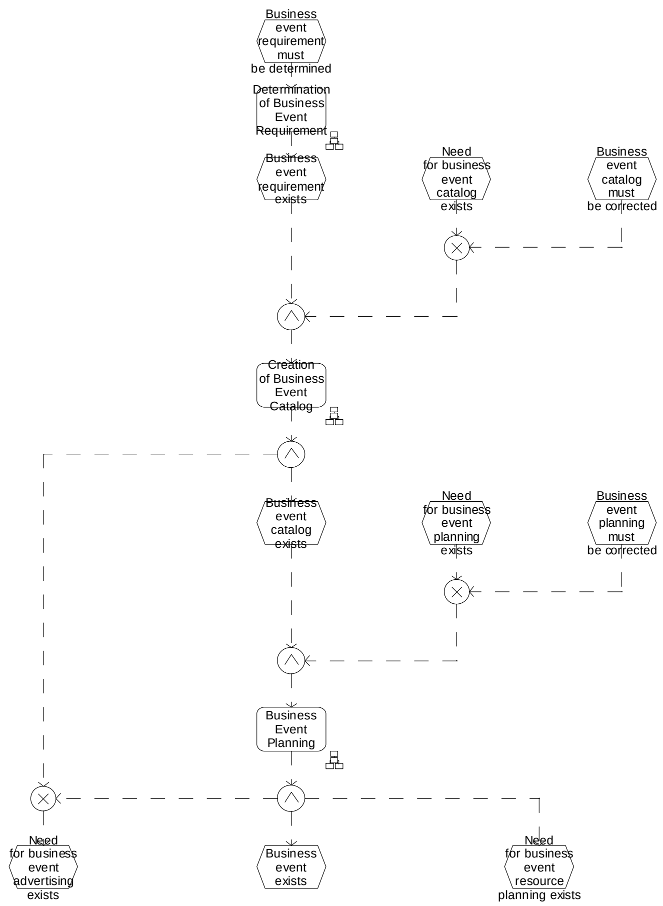


Figure B.55: Training and Event Management – Business Event Planning and Performance – Business Event Planning (reduced size 6, unsound)

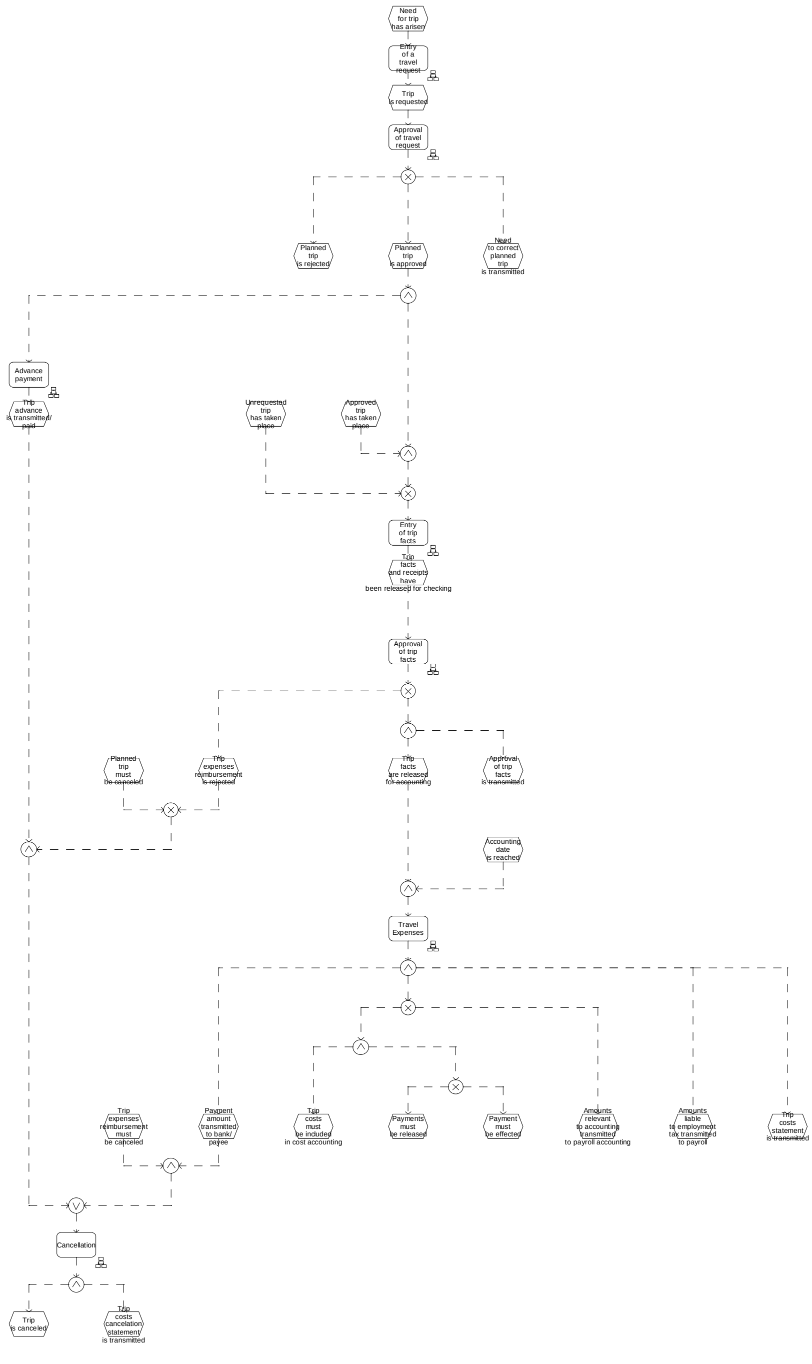


Figure B.56: Travel Management – Travel Expenses (reduced size 16, unsound)

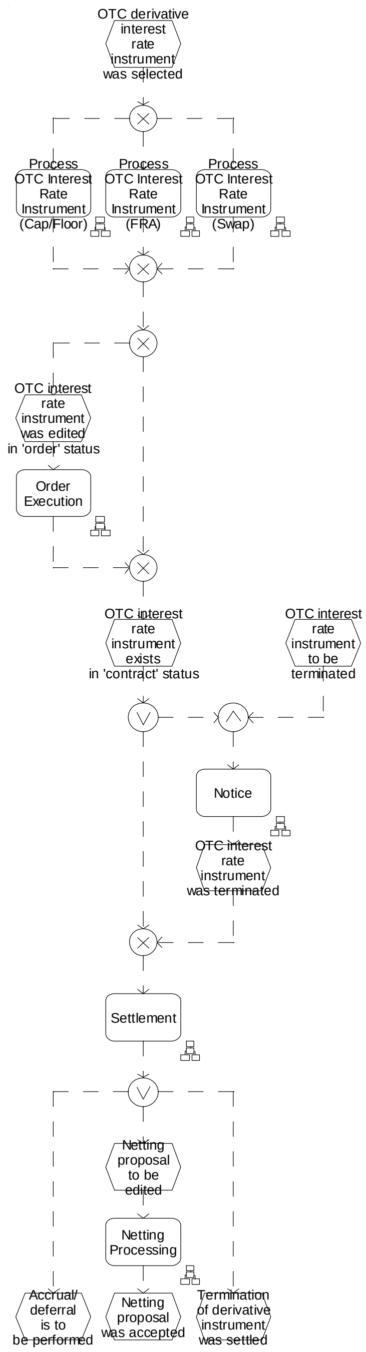


Figure B.57: Treasury – Process OTC Derivative Transactions (TR-DE) – Transaction Processing (reduced size 6, unsound)

Appendix C

Descriptive Statistics of Variables

This appendix gathers details of the statistical analysis. In particular, Section C.1 gives a tabular overview of the variables that were available for the statistical analysis. Section C.2 presents box plots that illustrate the empirical distribution of the variables disaggregated by the group of models. Section C.3 shows box plots for the different variables disaggregated by the variable *hasErrors*. Finally, Section C.5 contains the correlation tables between the variable *hasError* and the different metrics.

C.1 Definition of Variables

This section gives two tables that describe the variables that were available for the statistical analysis. Apart from the variable *countProM* and *hasErrors* all variable values were generated by *xoEPC*.

Table C.1: Variables of the analysis table (first part)

Variable name	**Description**
Group	Number of the EPC collection group
Filename	Name of the ARIS XML file
Model ID	ID of the EPC model
Duration	Processing time in milliseconds
Path	Path of the EPC within the model hierarchy of the file
Name	Name of the EPC model
Error	Value 1 if xoEPC found errors, otherwise 0
Reduced	Value 1 if the EPC was reduced completely, otherwise 0
Restsize	Size in nodes of the reduced EPC
Interpretable	Value 1 if relaxed syntactically correct, otherwise 0
Syntax	List of syntax error descriptions
N	Number of nodes
C	Number of connectors
E	Number of events
Es	Number of start events
Ee	Number of end events
F	Number of functions
AND	Number of AND-connectors
XOR	Number of XOR-connectors
OR	Number of OR-connectors
ANDj	Number of AND-joins
XORj	Number of XOR-joins
ORj	Number of OR-joins
ANDs	Number of AND-splits
XORs	Number of XOR-splits
ORs	Number of OR-splits
A	Number of arcs
diameter	Diameter

Table C.2: Variables of the analysis table (second part)

Variable name	Description
Density	Density metric
CNC	Coefficient of connectivity
AvCDegree	Average connector degree
MaxCDegree	Maximum connector degree
Separability	Separability ratio
Sequentiality	Sequentiality ratio
Structuredness	Structuredness ratio
Depth	Depth
MM	Connector mismatch
cHeterogeneity	Connector heterogeneity
CFC	Control flow complexity
CYC	Cyclicity
tokenSplit	Token split
rsequence	Number of trivial construct rule application
rblock	Number of structured block rule application
rloop	Number of structured loop rule application
rstartend	Number of structured start and end rule application
rjump	Number of unstructured start and end rule application
rdelta	Number of delta rule application
rprism	Number of prism rule application
rmerge	Number of merge rule application
rxoronly	Number of nodes deleted by homogeneous rule application
countblock	Number of structured block errors
countloop	Number of structured loop errors
countdelta	Number of delta errors
countprism	Number of prism errors
countsplitend	Number of unstructured start and end errors
countProM	Value 1 if errors detected by ProM, otherwise 0
hasErrors	Value 1 if errors, otherwise 0

C.2 Box plots filtered by model group

This section shows box plots of each variable disaggregated by the group of models. The boolean variables *Error*, *Reduced*, *Interpretable*, *countProM*, and *hasError* are not included since box plots are made for interval scale.

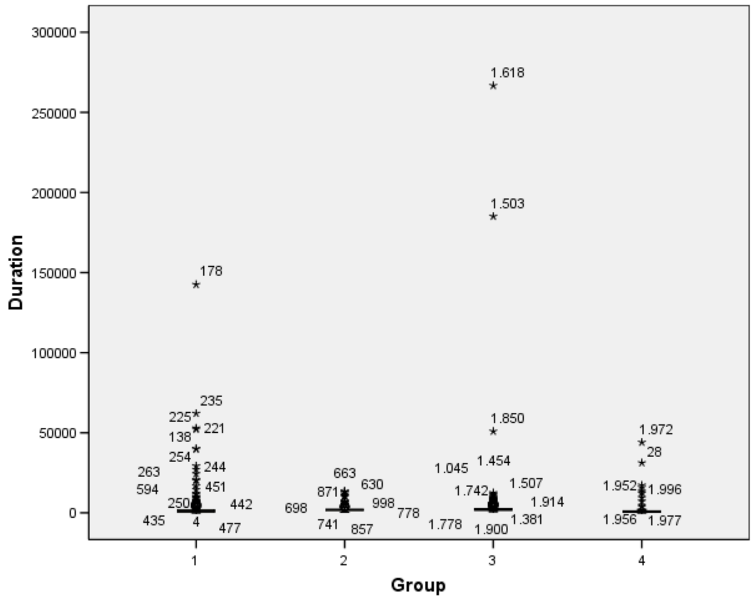


Figure C.1: Box plot for duration by group

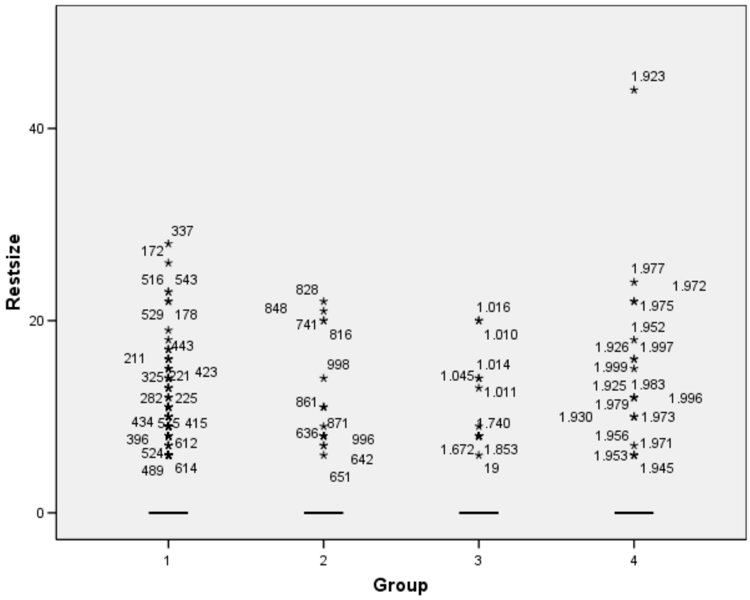


Figure C.2: Box plot for restsize by group

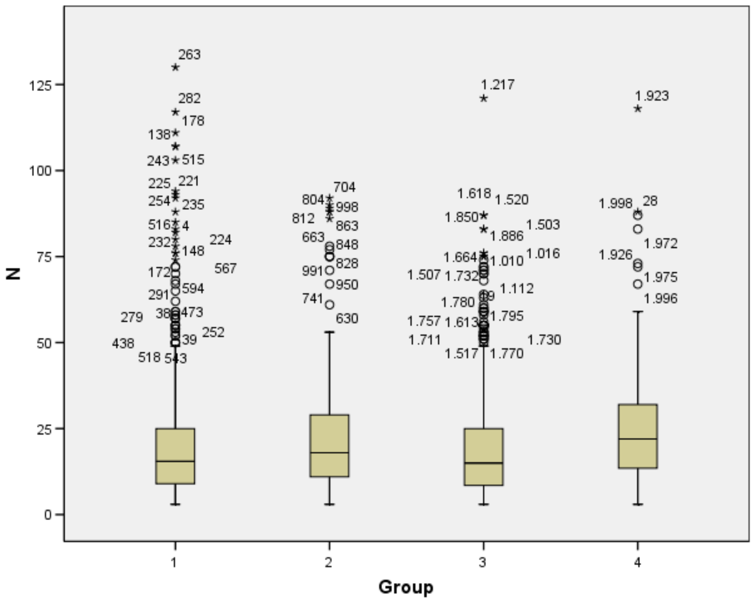


Figure C.3: Box plot for nodes N by group

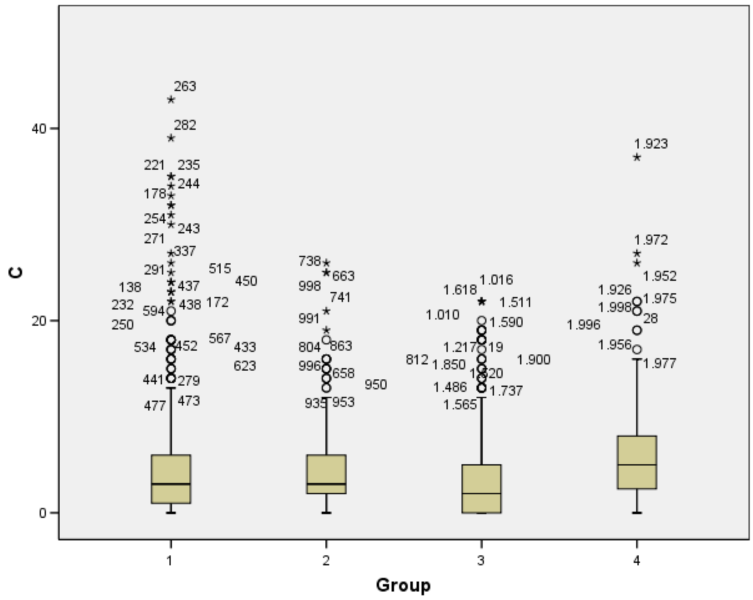


Figure C.4: Box plot for connectors C by group

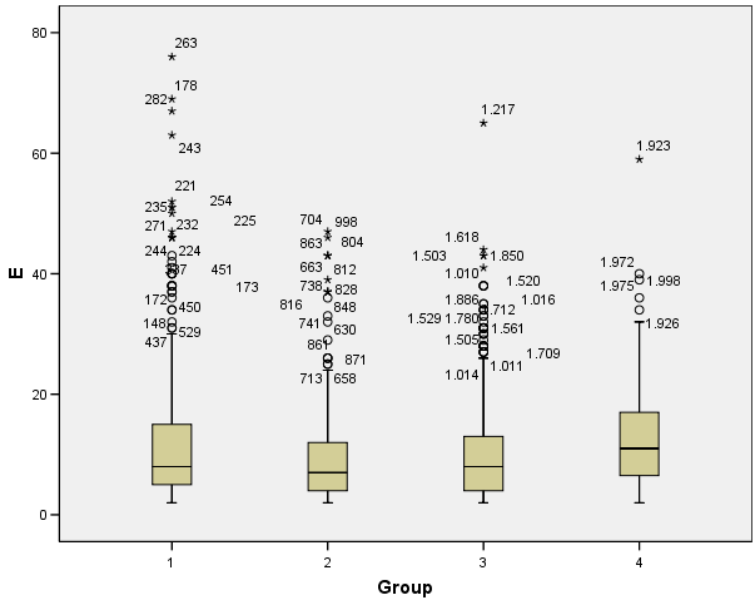


Figure C.5: Box plot for events E by group

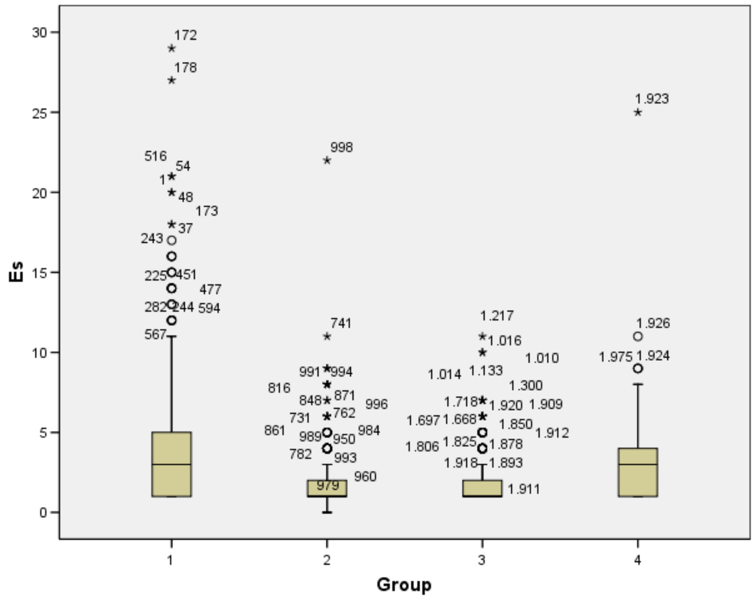


Figure C.6: Box plot for start events Es by group

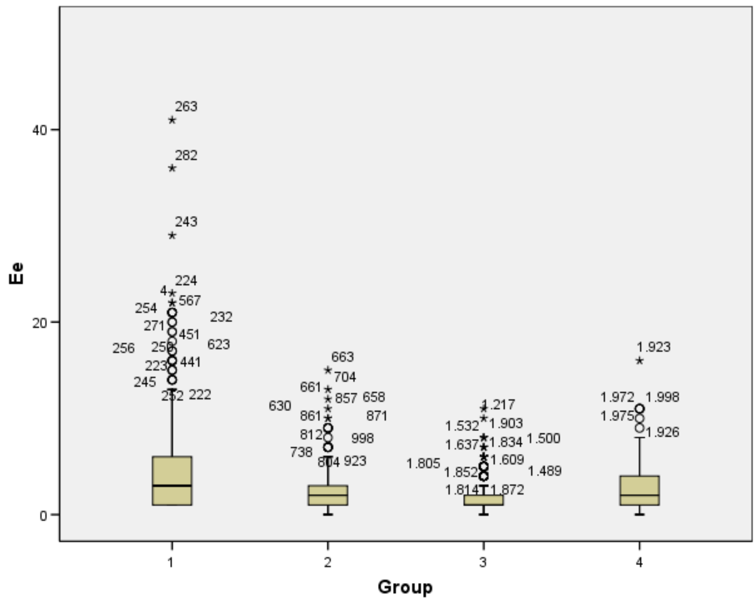


Figure C.7: Box plot for end events Ee by group

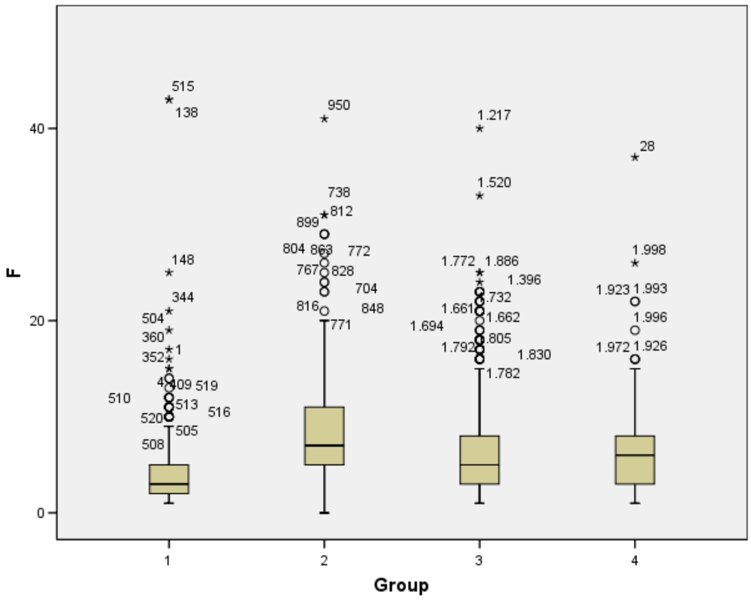


Figure C.8: Box plot for functions F by group

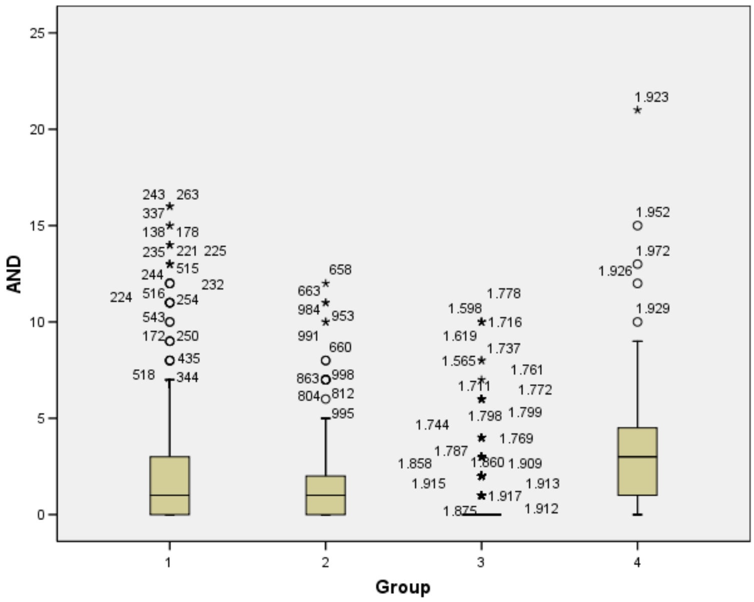


Figure C.9: Box plot for AND-connectors by group

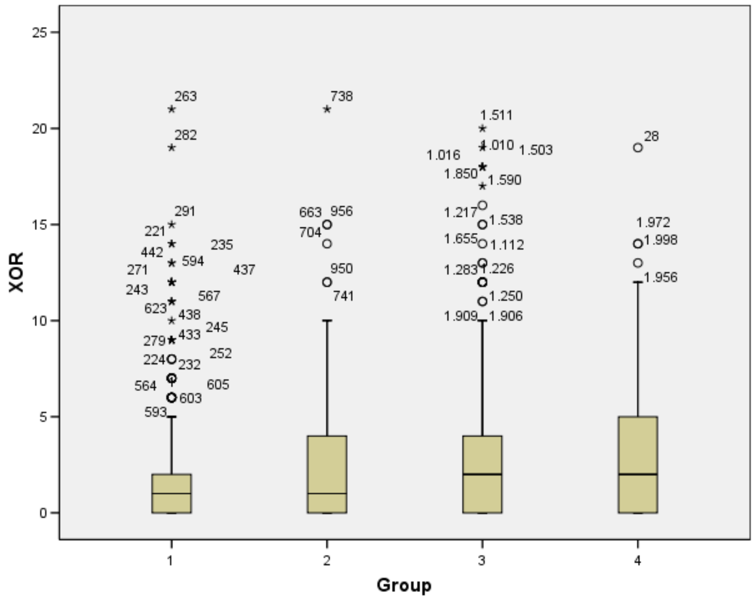


Figure C.10: Box plot for XOR-connectors by group

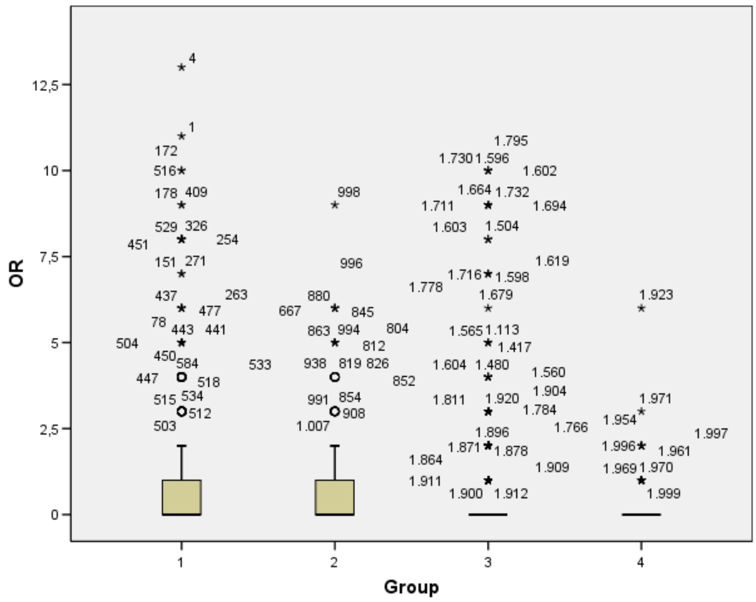


Figure C.11: Box plot for OR-connectors by group

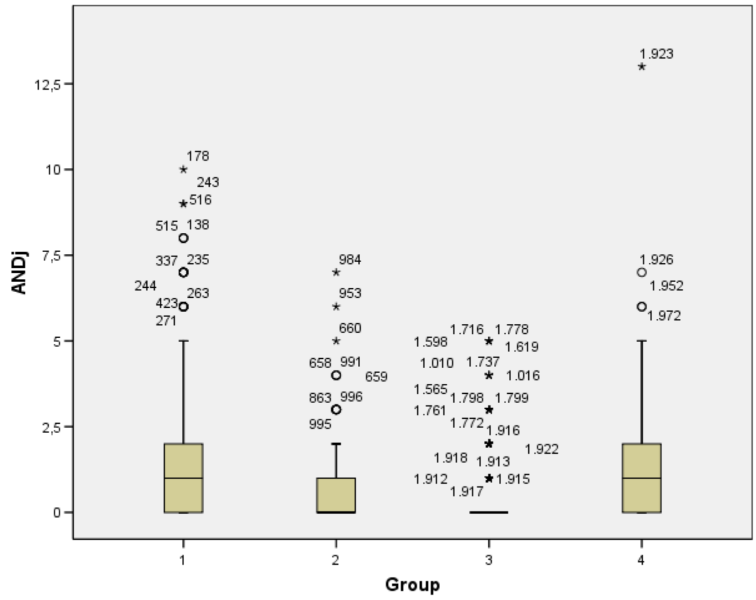


Figure C.12: Box plot for AND-joins by group

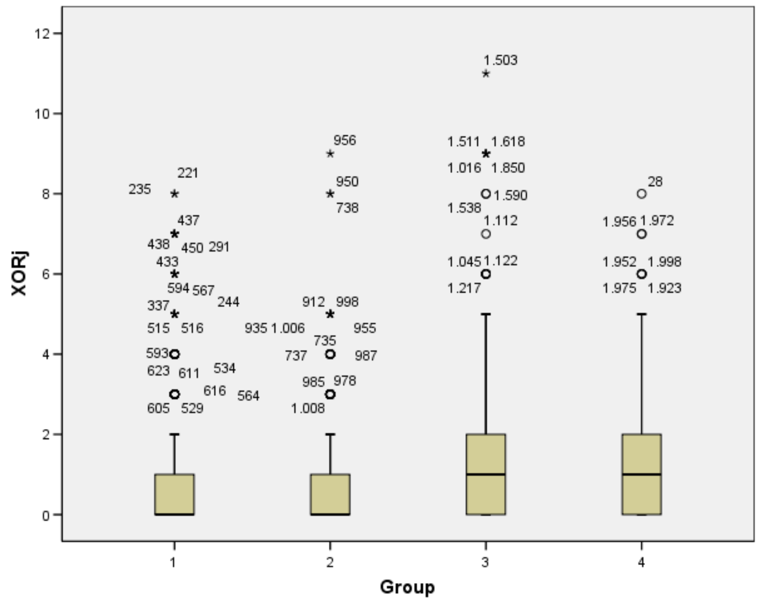


Figure C.13: Box plot for XOR-joins by group

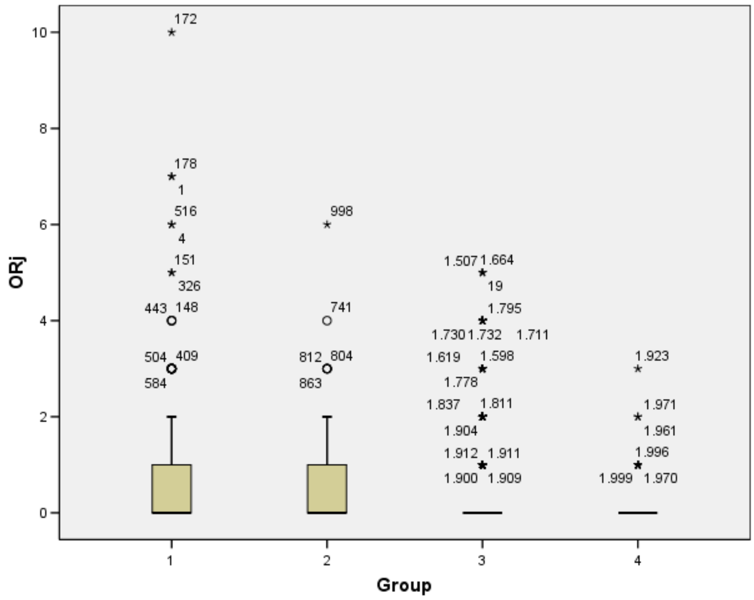


Figure C.14: Box plot for OR-joins by group

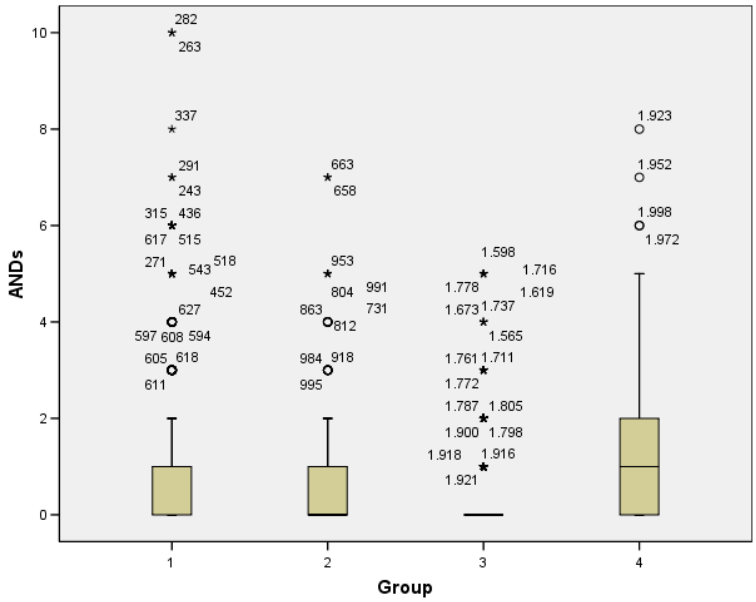


Figure C.15: Box plot for AND-splits by group

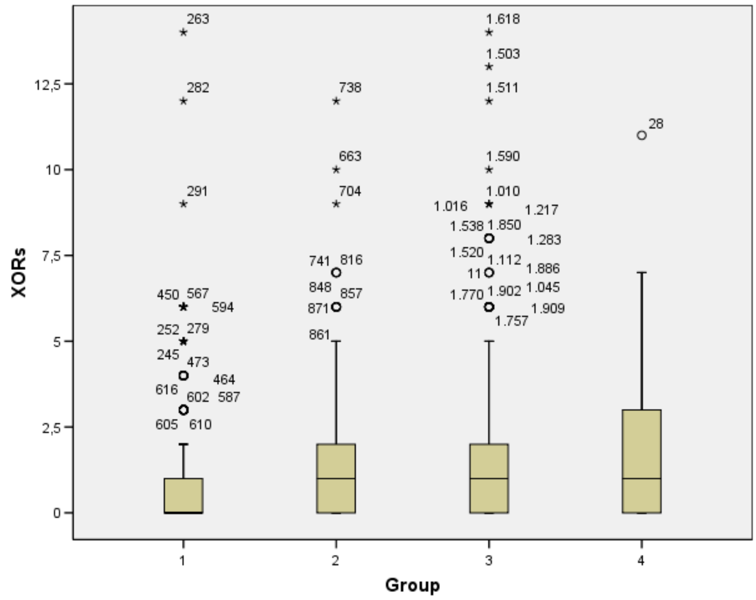


Figure C.16: Box plot for XOR-splits by group

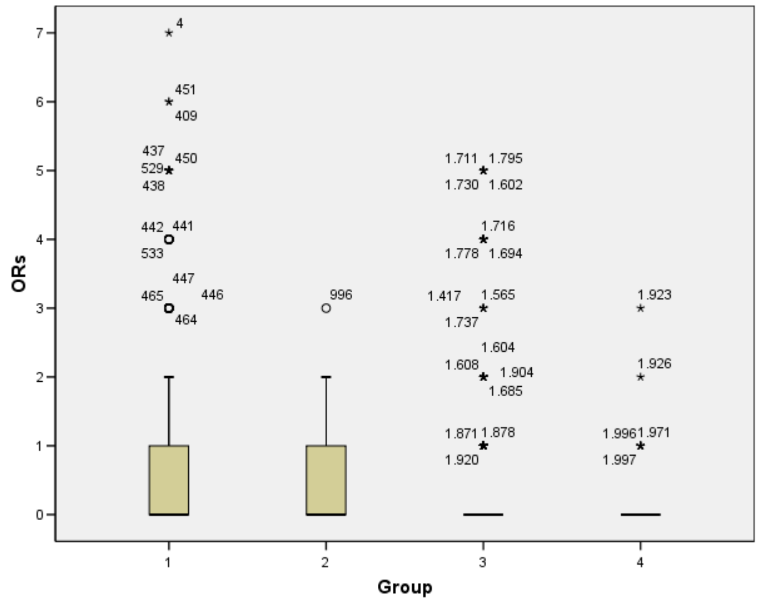


Figure C.17: Box plot for OR-splits by group

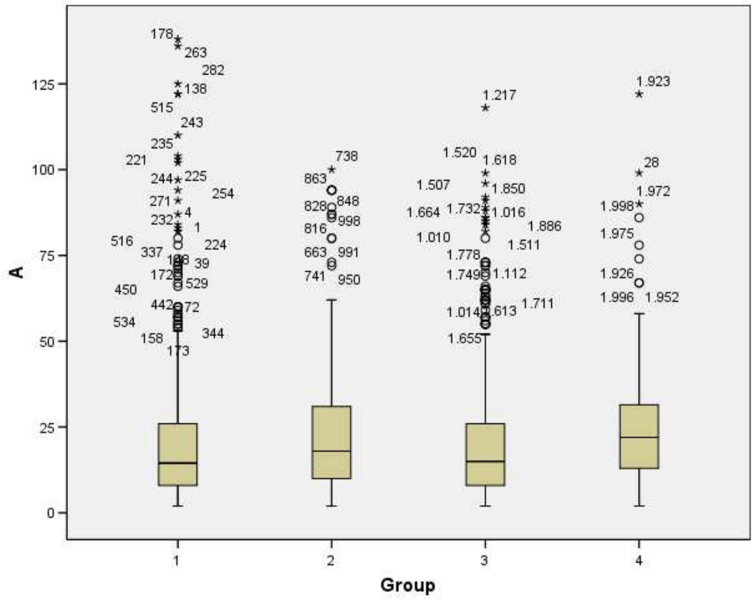


Figure C.18: Box plot for arcs A by group

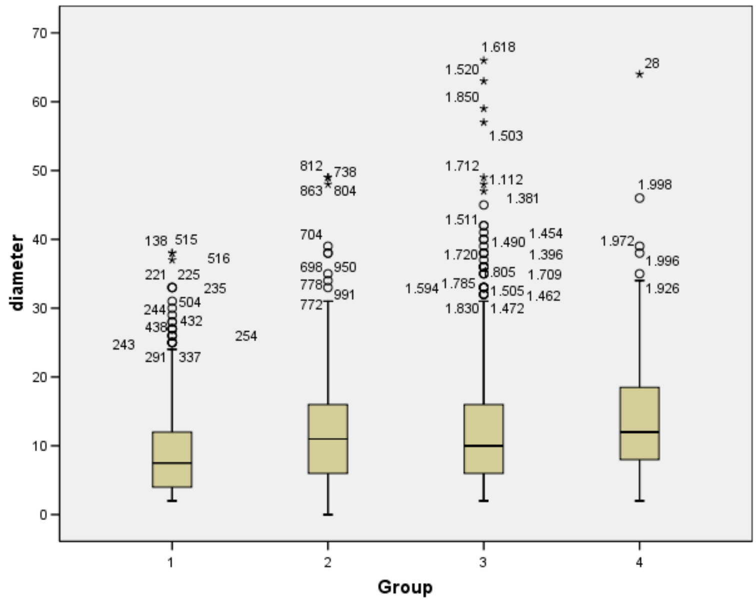


Figure C.19: Box plot for diameter by group

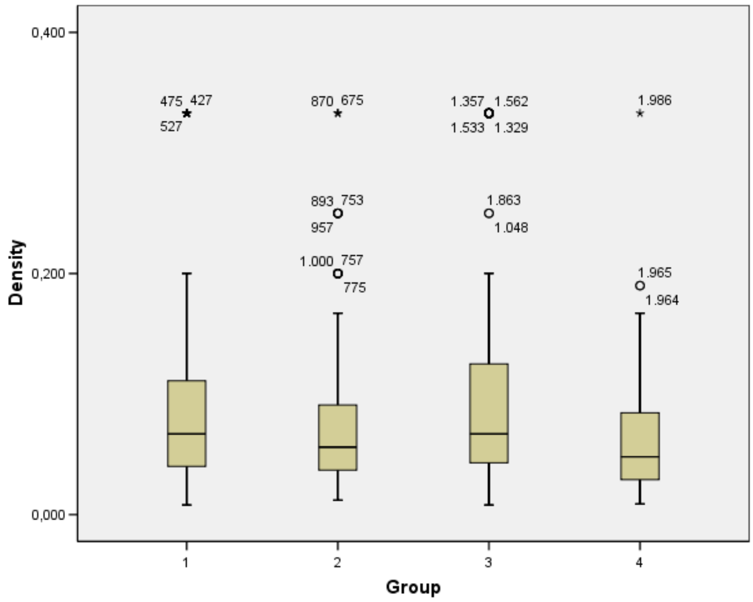


Figure C.20: Box plot for density by group

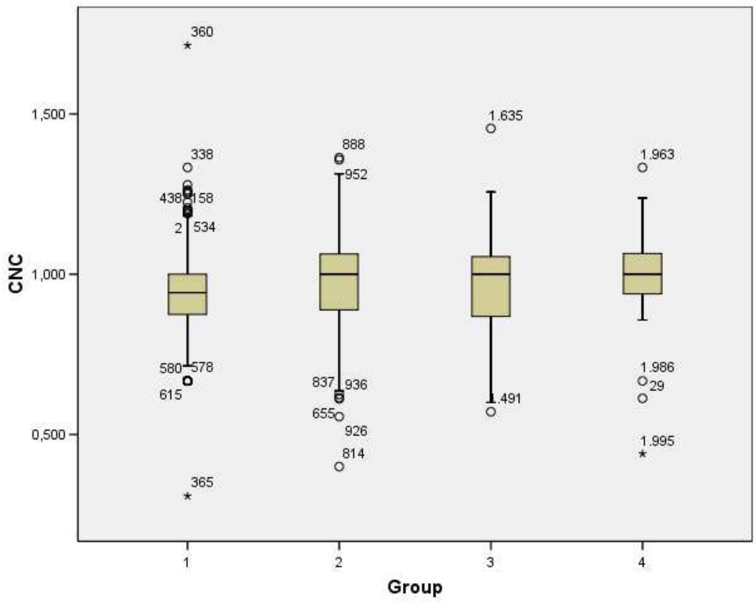


Figure C.21: Box plot for coefficient of connectivity CNC by group

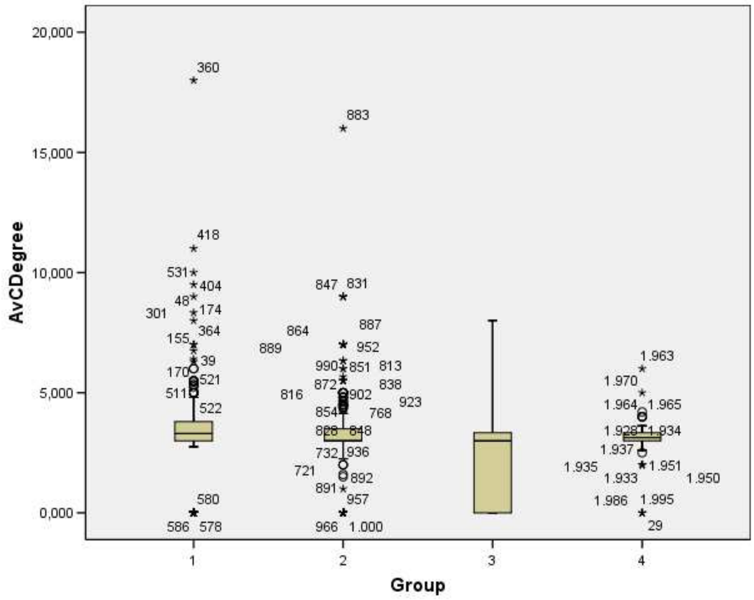


Figure C.22: Box plot for average connector degree by group

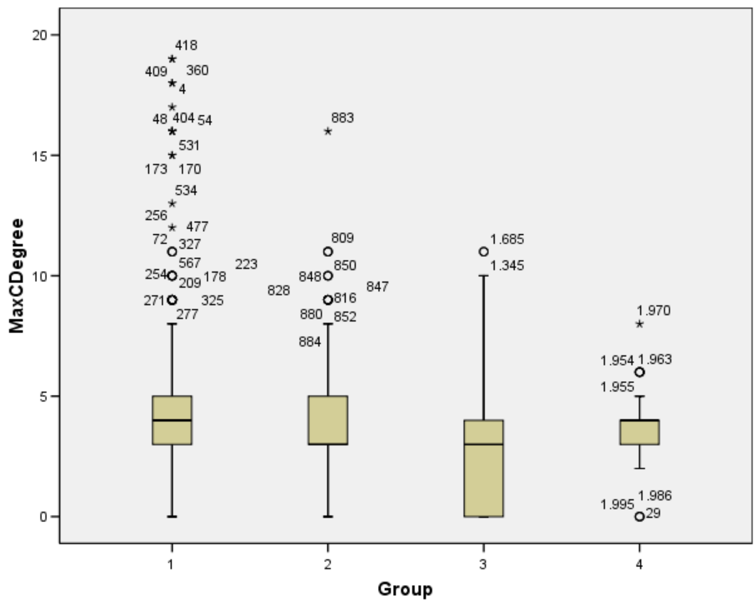


Figure C.23: Box plot for maximum connector degree by group

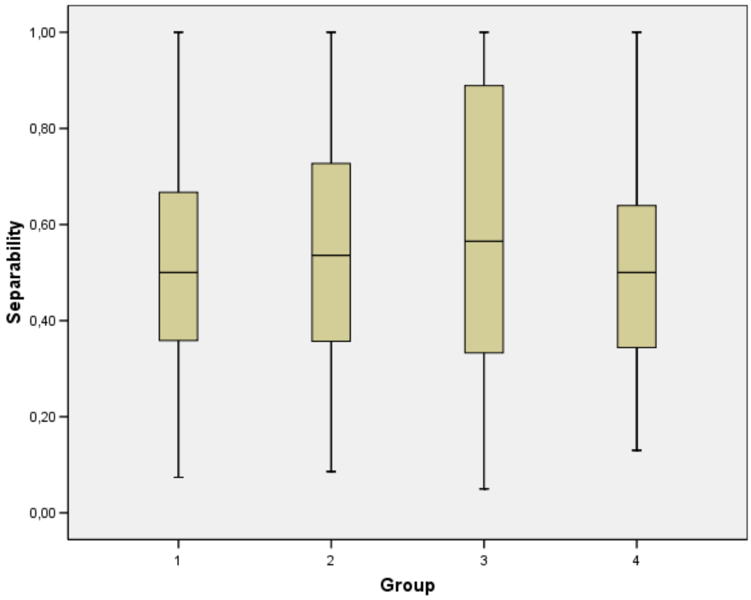


Figure C.24: Box plot for separability by group

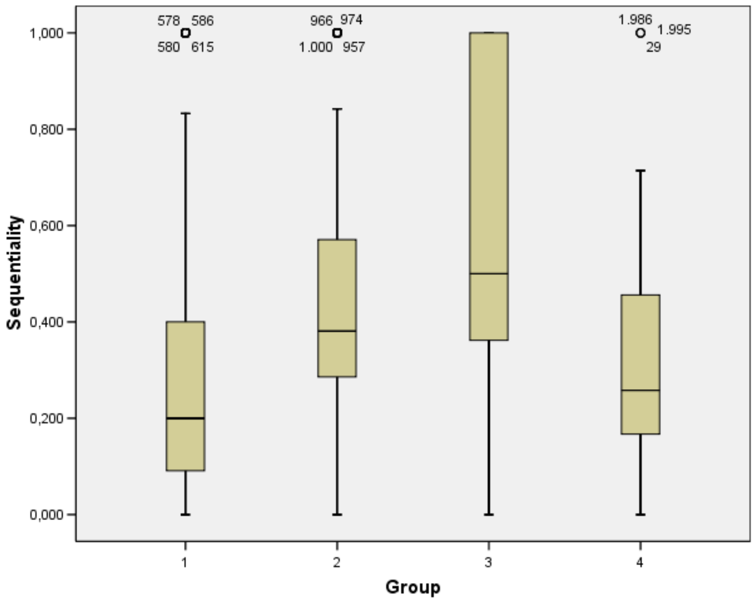


Figure C.25: Box plot for sequentiality by group

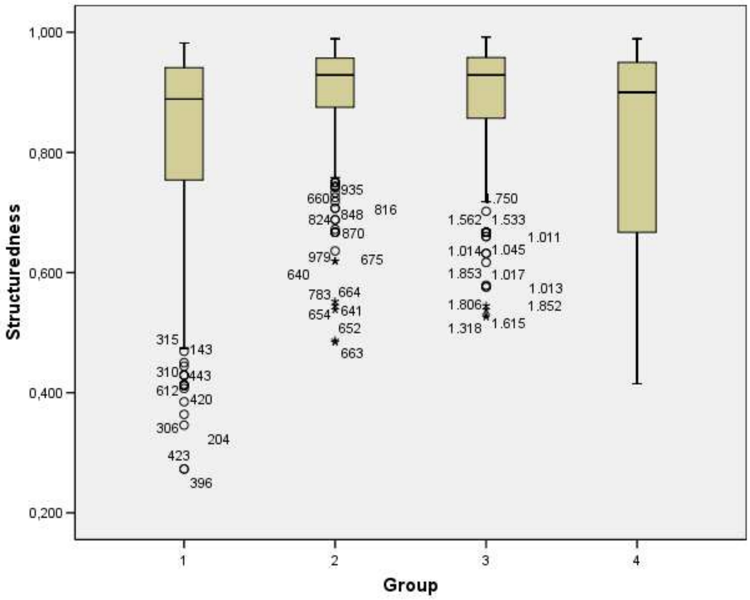


Figure C.26: Box plot for structuredness by group

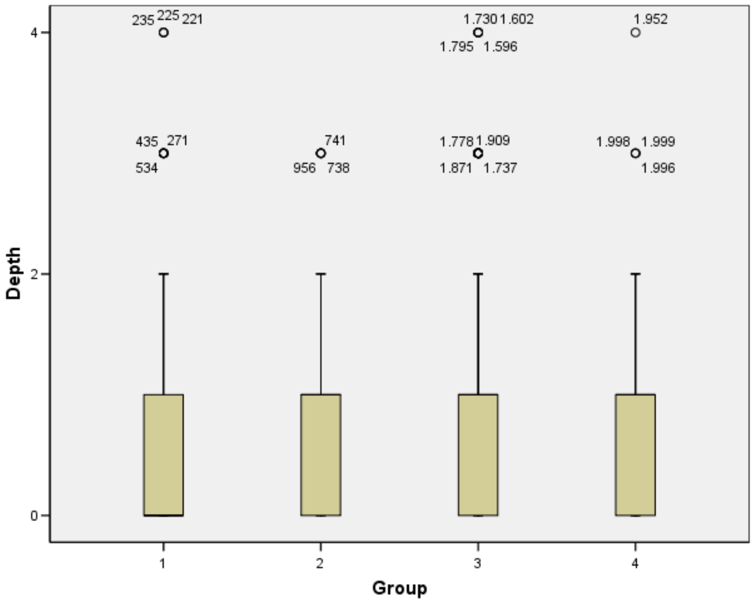


Figure C.27: Box plot for depth by group

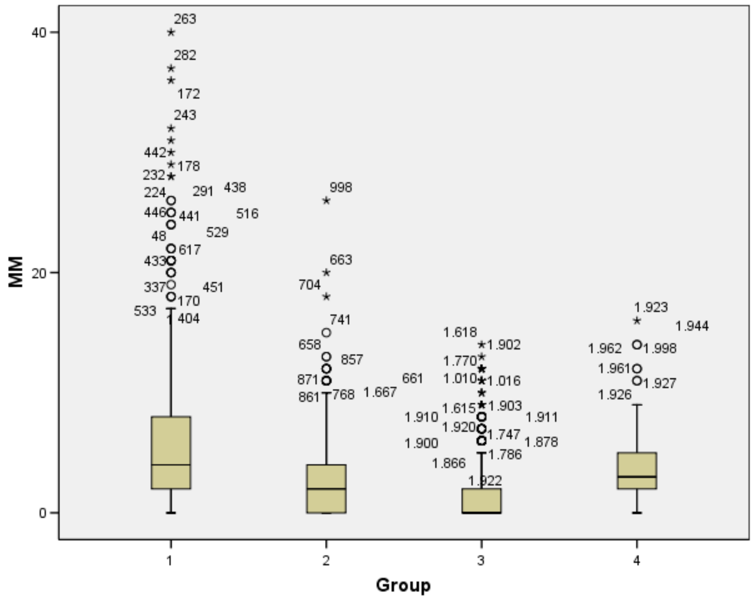


Figure C.28: Box plot for connector mismatch MM by group

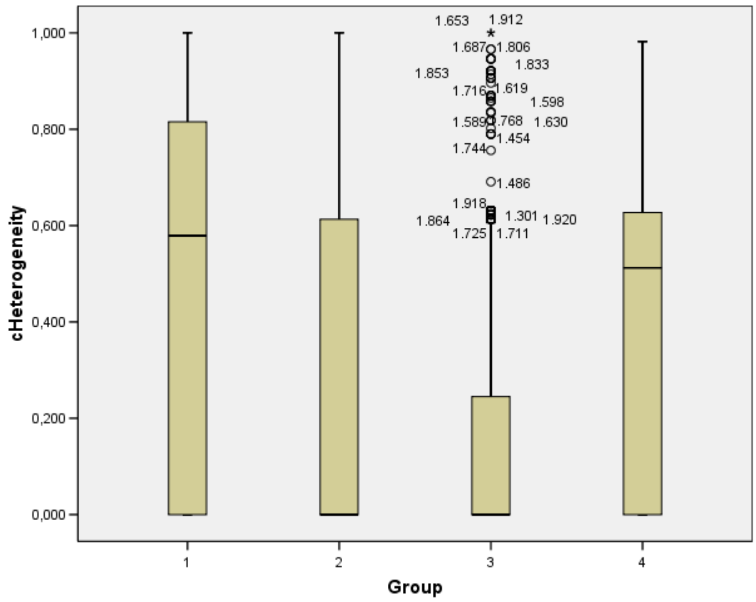


Figure C.29: Box plot for connector heterogeneity by group

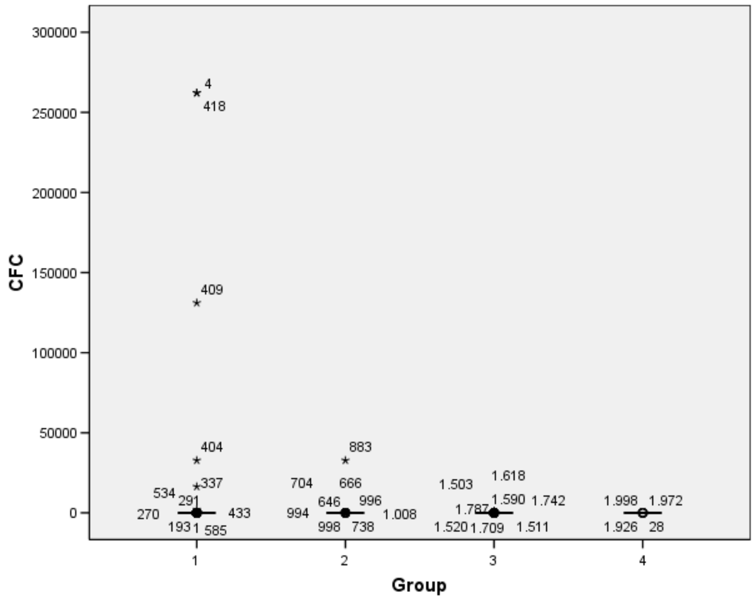


Figure C.30: Box plot for control flow complexity CFC by group

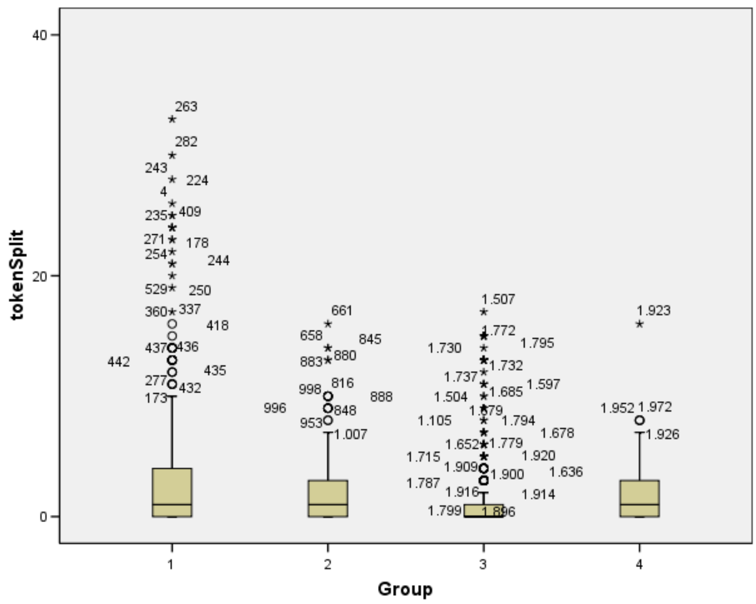


Figure C.31: Box plot for token split by group

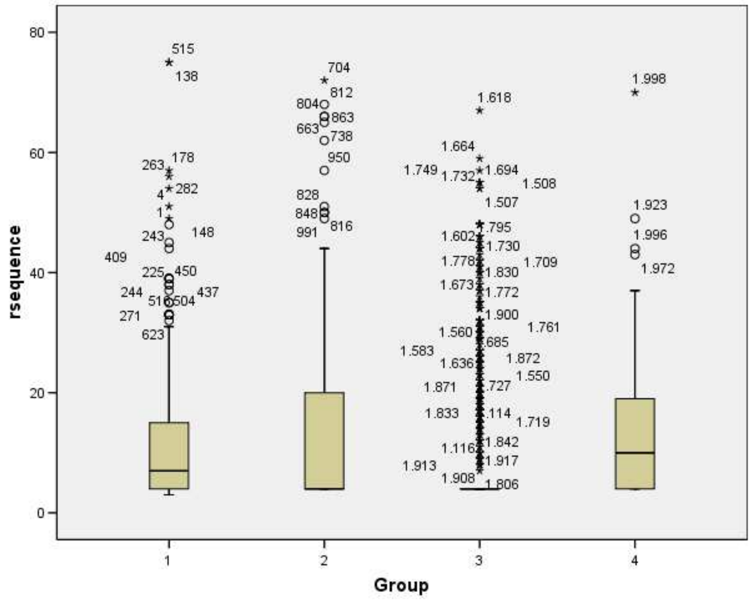


Figure C.32: Box plot for trivial construct rule application by group

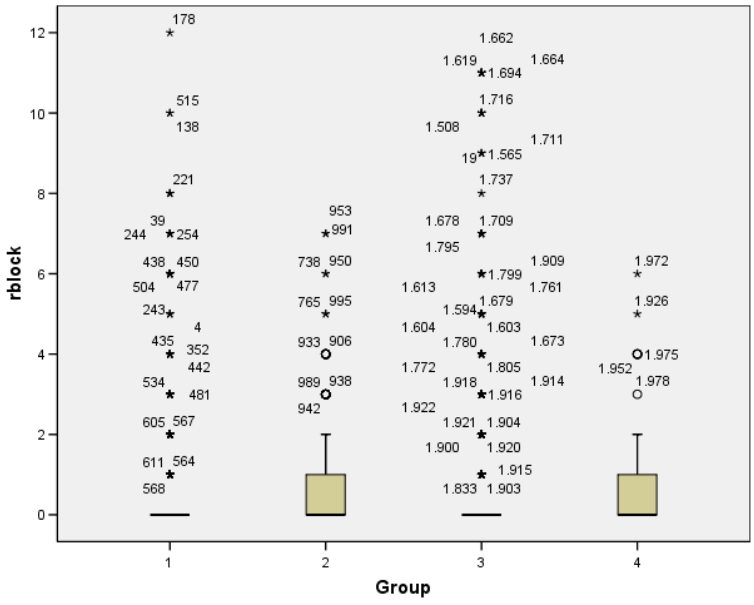


Figure C.33: Box plot for structured block rule application by group

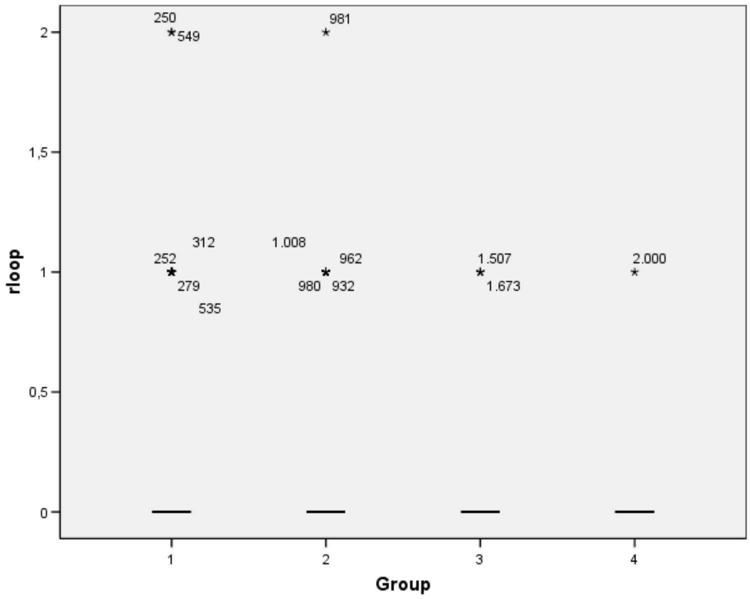


Figure C.34: Box plot for structured loop rule application by group

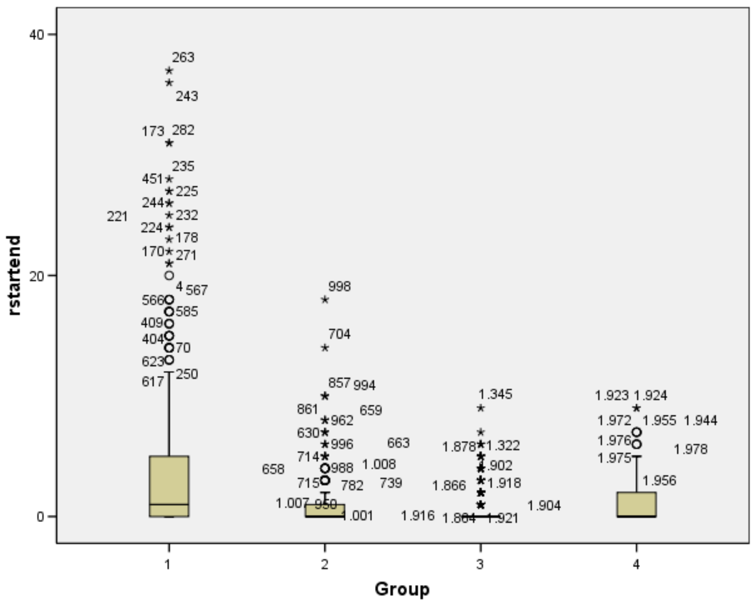


Figure C.35: Box plot for structured start and end rule application by group

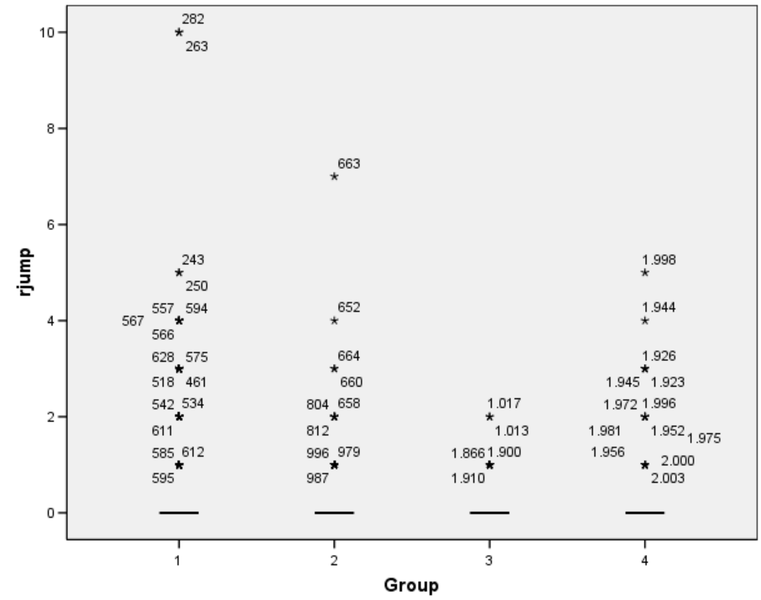


Figure C.36: Box plot for unstructured start and end rule application by group

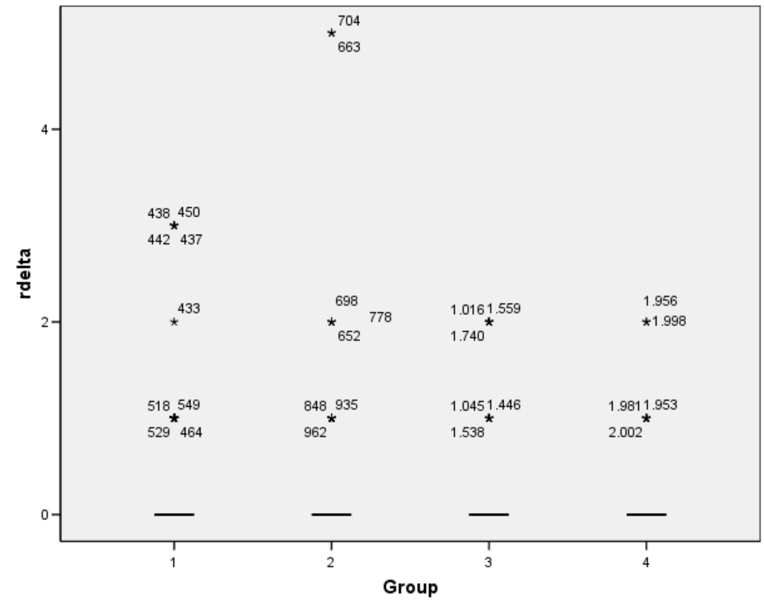


Figure C.37: Box plot for delta rule application by group

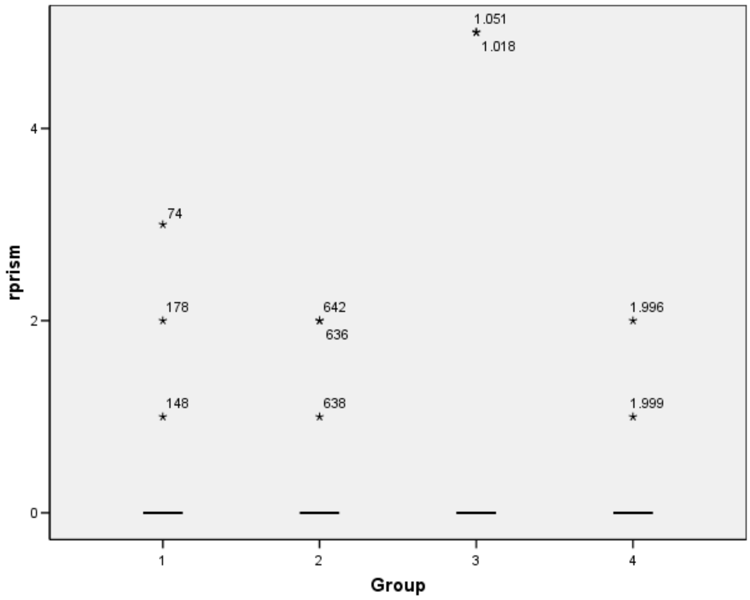


Figure C.38: Box plot for prism rule application by group

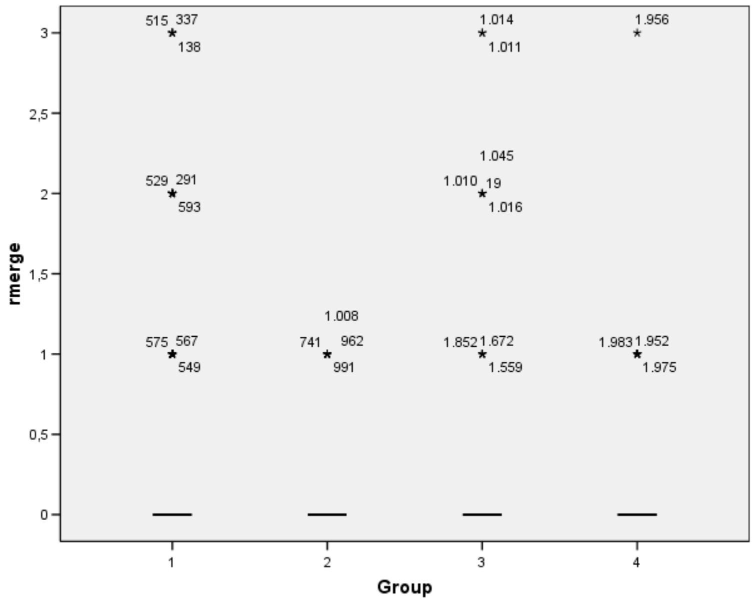


Figure C.39: Box plot for connector merge rule application by group

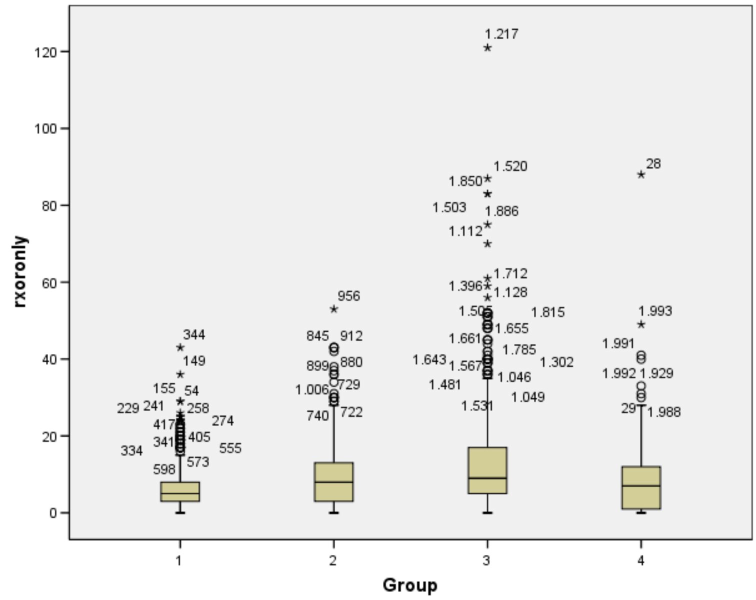


Figure C.40: Box plot for homogeneous rule application by group

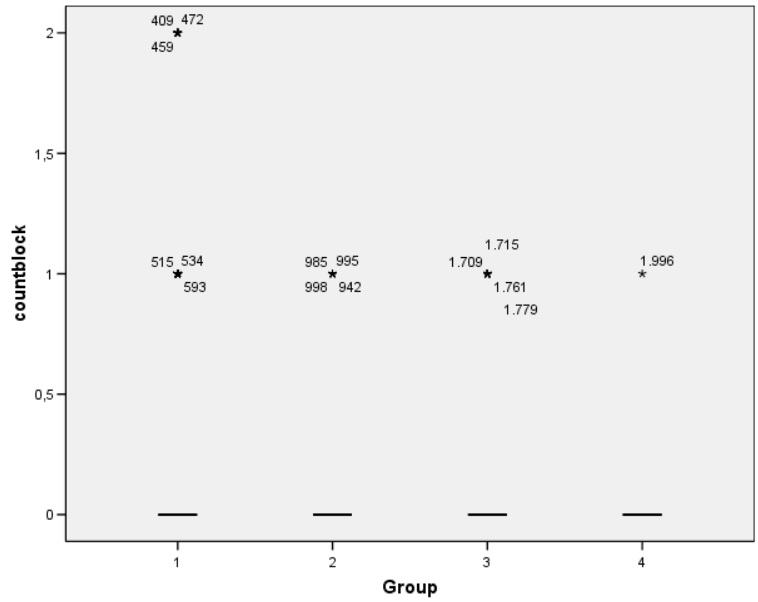


Figure C.41: Box plot for structured block errors by group

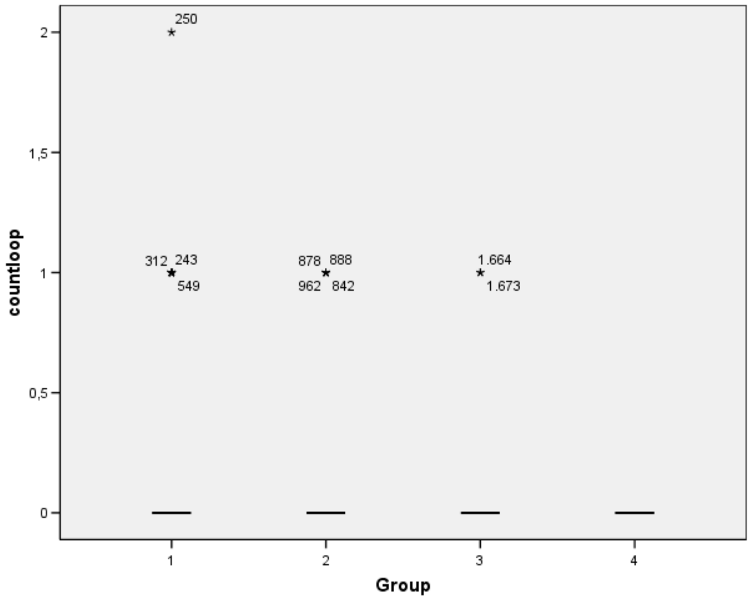


Figure C.42: Box plot for structured loop errors by group

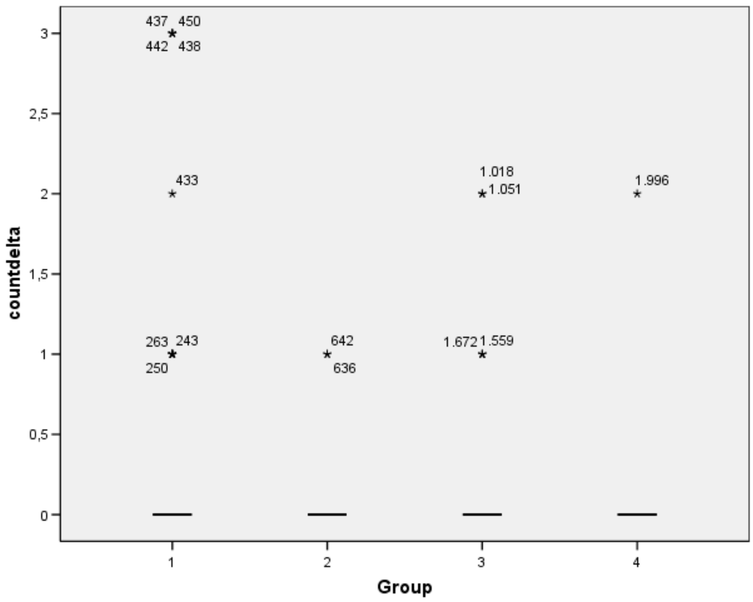


Figure C.43: Box plot for delta errors by group

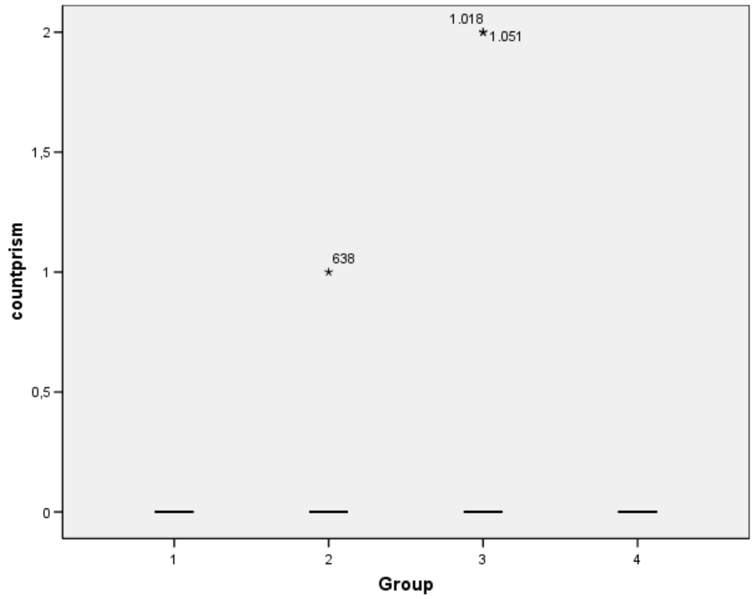


Figure C.44: Box plot for prism errors by group

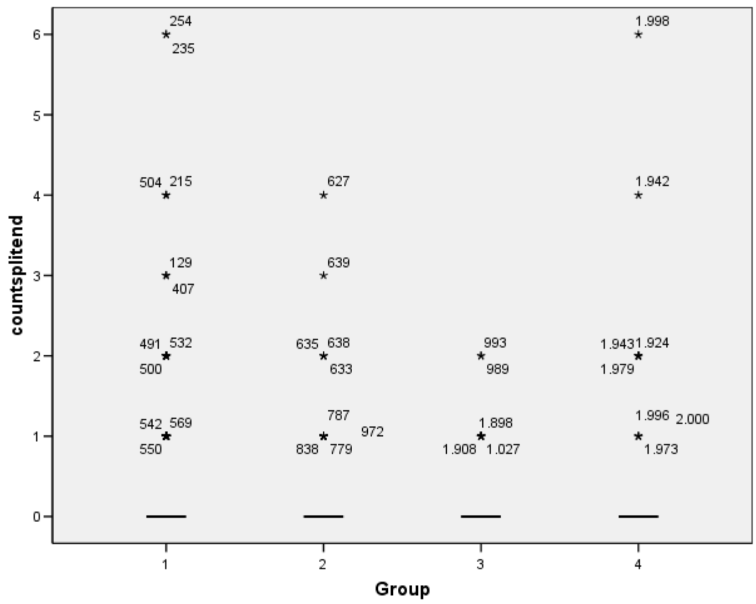


Figure C.45: Box plot for TODO unstructured start and end errors by group

C.3 Box plots filtered by error

This section shows box plots of each variable disaggregated by the variable *hasErrors*. The boolean variables *Error*, *Reduced*, *Interpretable*, and *countProM* are not included since box plots are made for interval scale.

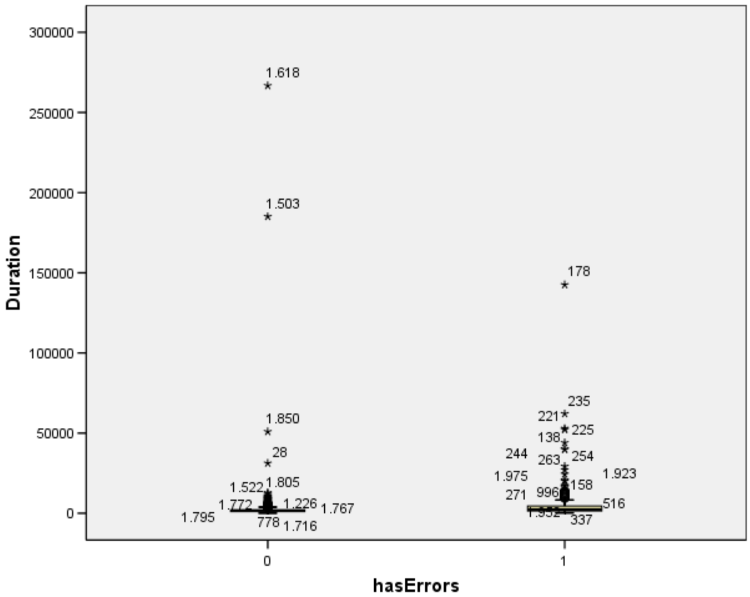


Figure C.46: Box plot for duration by error

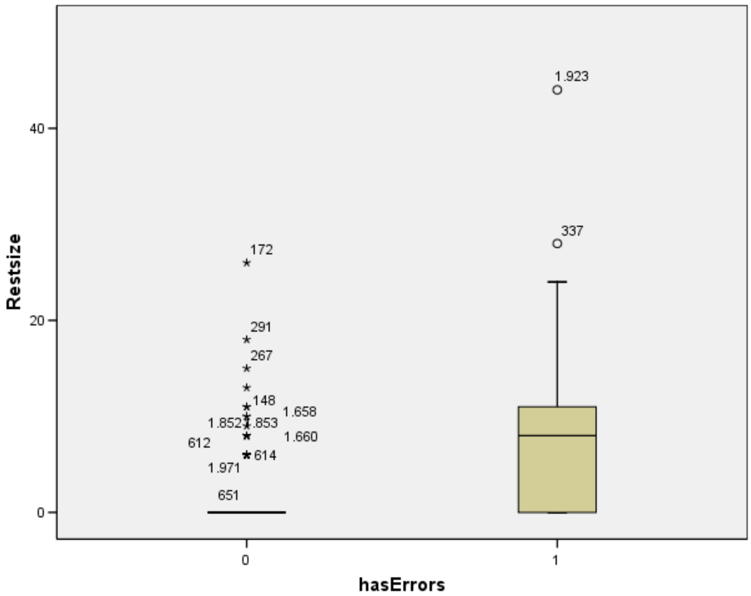


Figure C.47: Box plot for restsize by error

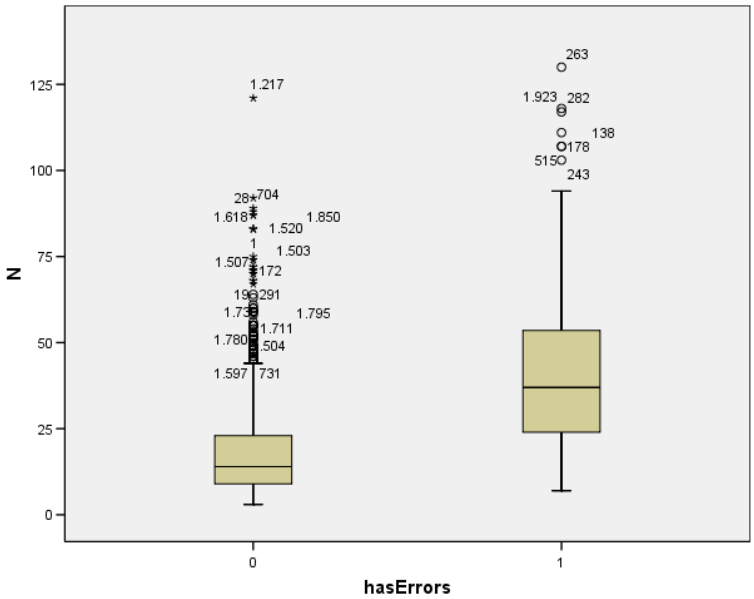


Figure C.48: Box plot for nodes N by error

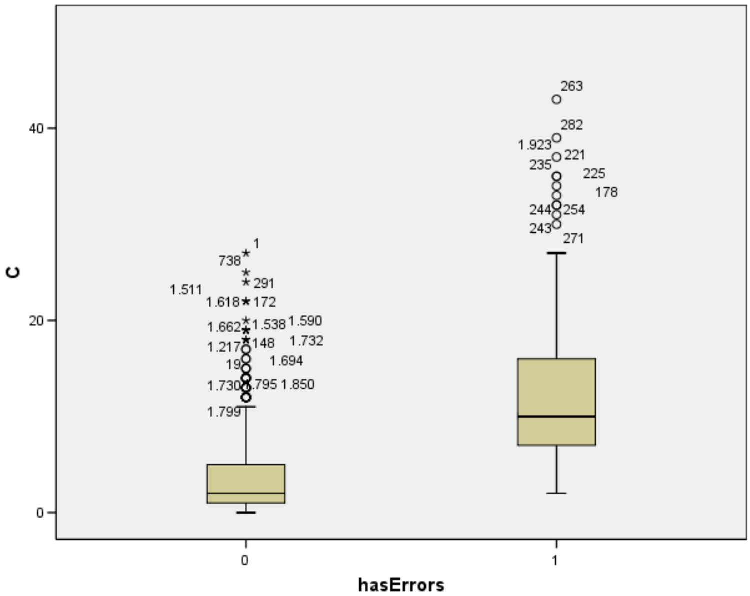


Figure C.49: Box plot for connectors C by error

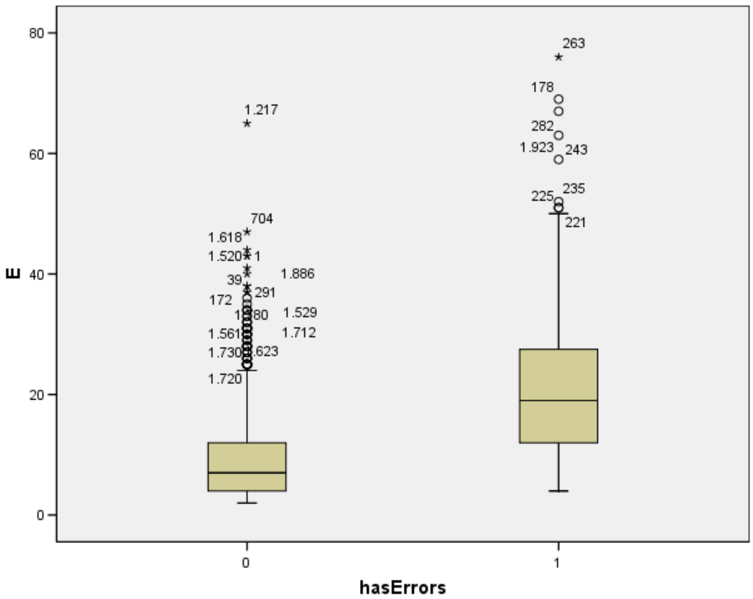


Figure C.50: Box plot for events E by error

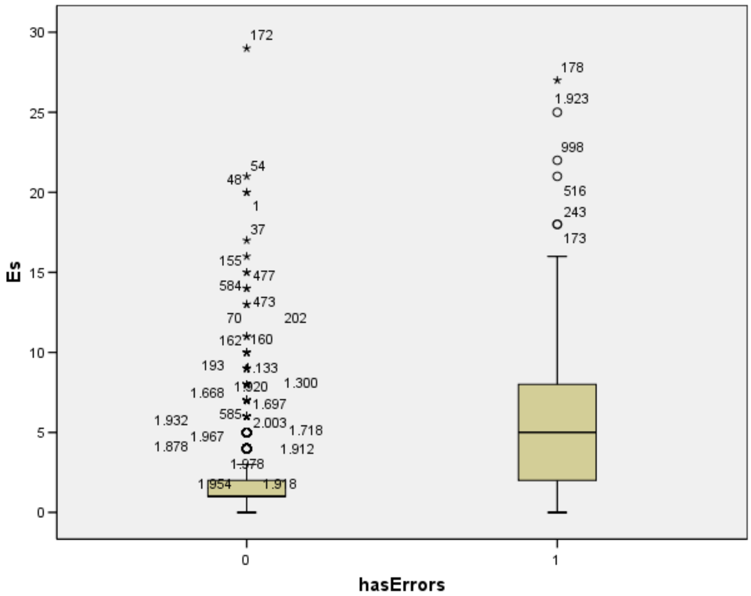


Figure C.51: Box plot for start events Es by error

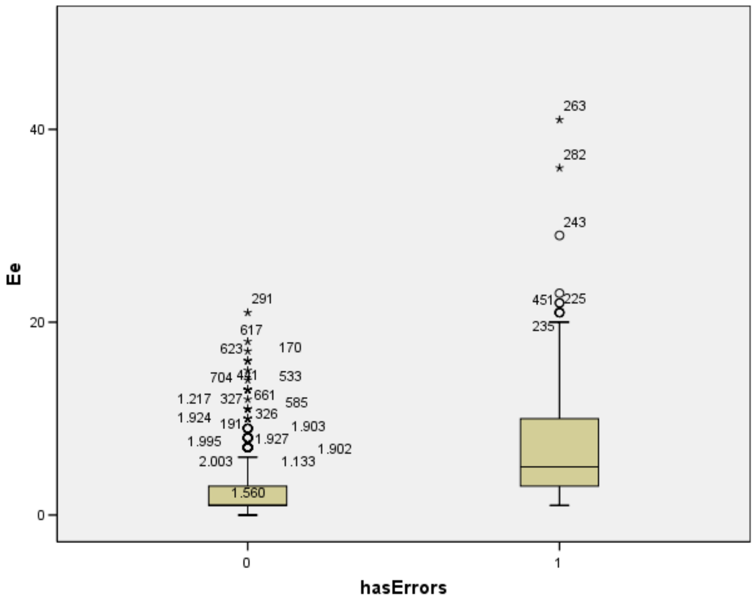


Figure C.52: Box plot for end events Ee by error

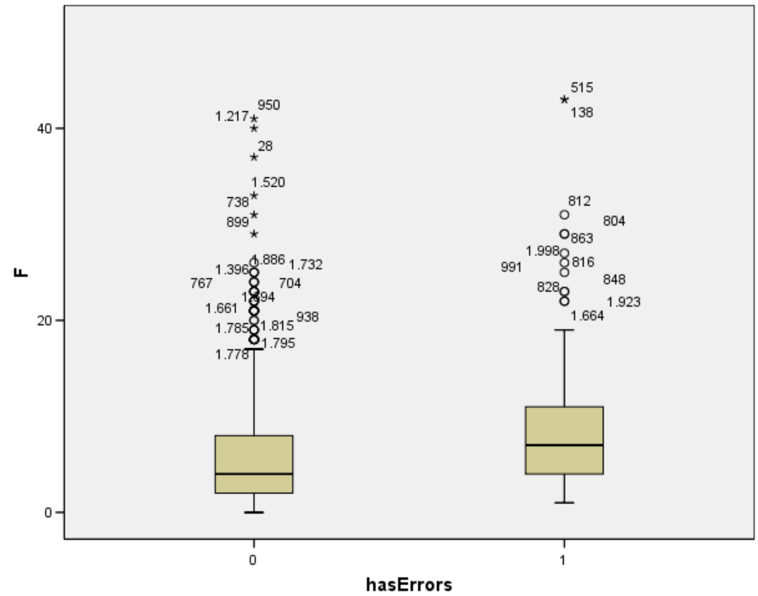


Figure C.53: Box plot for functions F by error

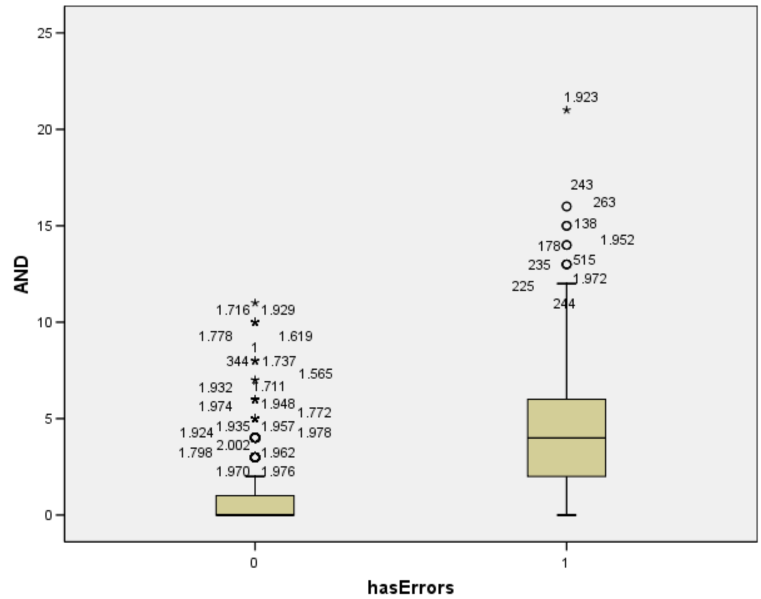


Figure C.54: Box plot for AND-connectors by error

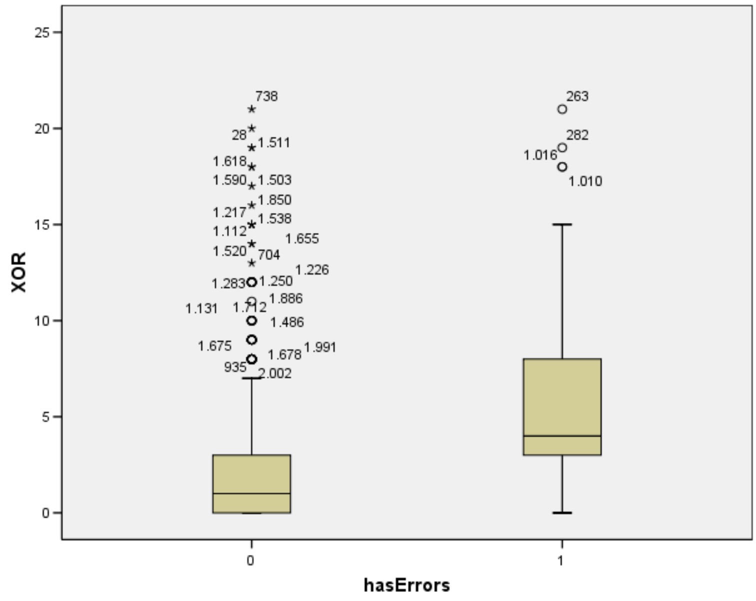


Figure C.55: Box plot for XOR-connectors by error

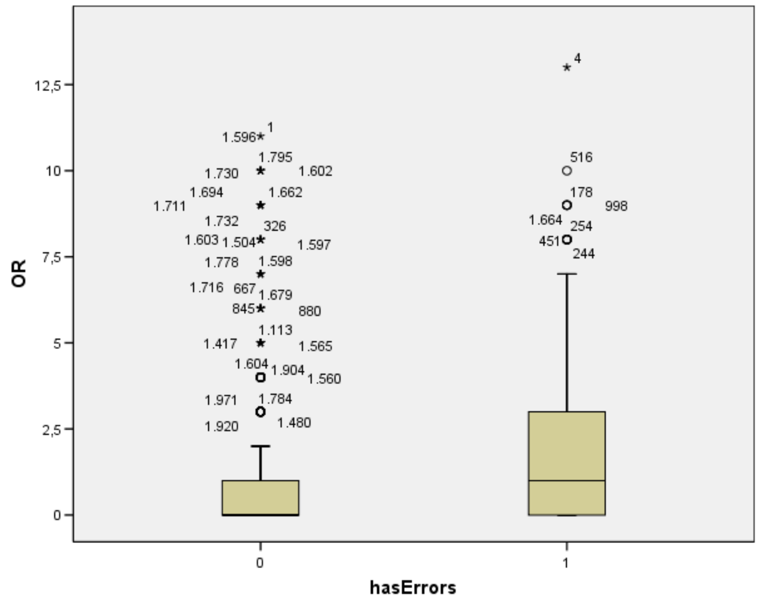


Figure C.56: Box plot for OR-connectors by error

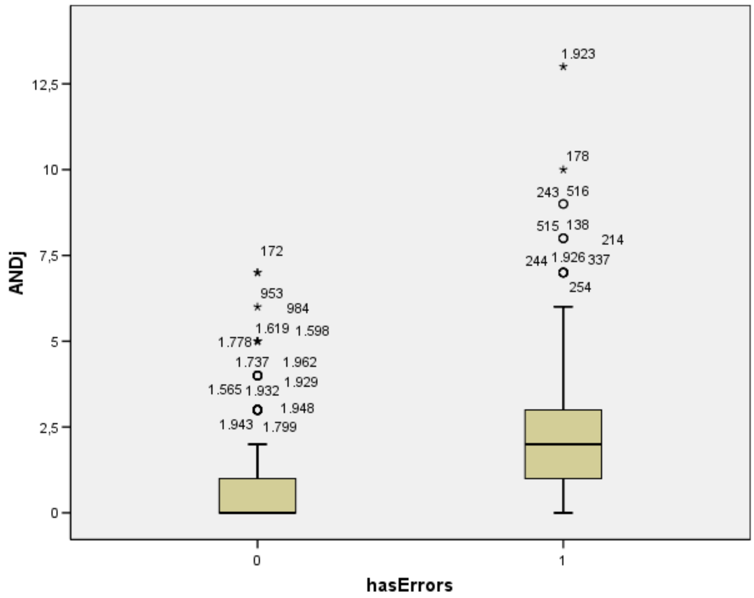


Figure C.57: Box plot for AND-joins by error

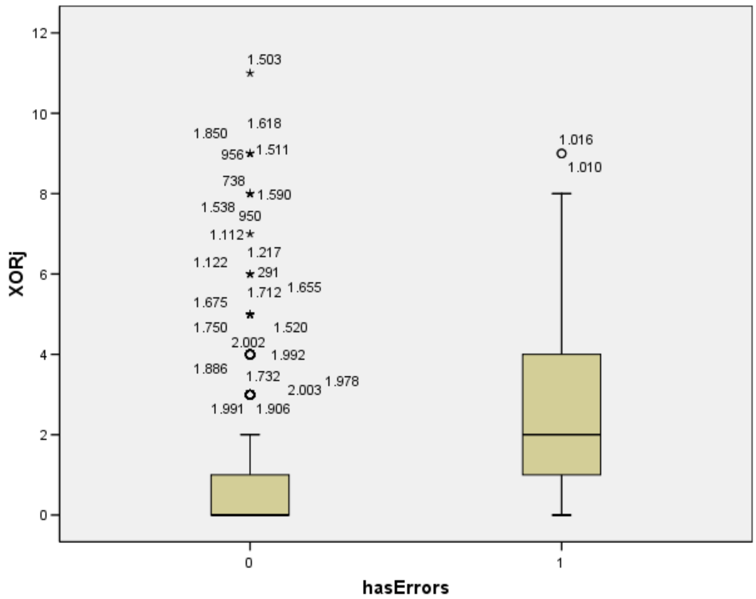


Figure C.58: Box plot for XOR-joins by error

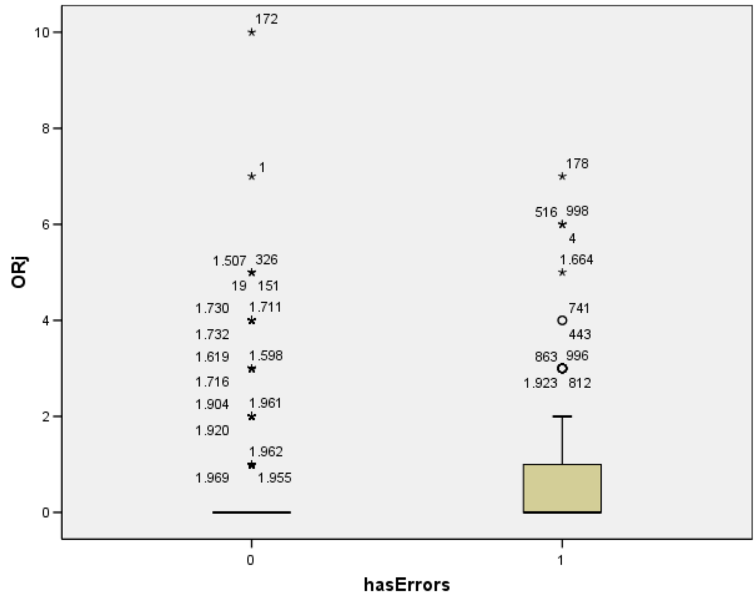


Figure C.59: Box plot for OR-joins by error

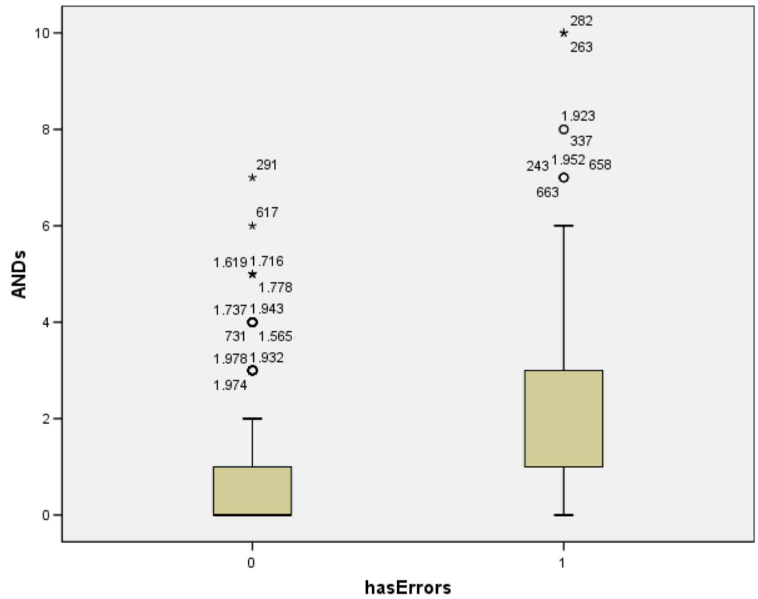


Figure C.60: Box plot for AND-splits by error

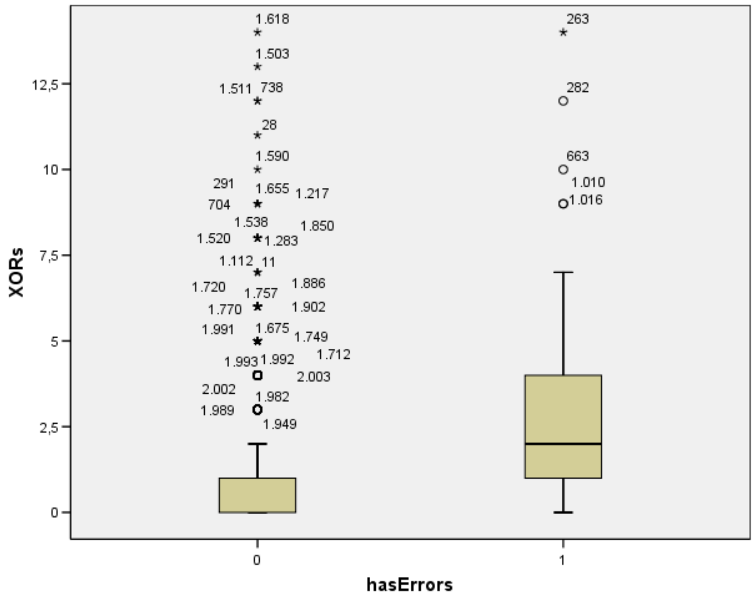


Figure C.61: Box plot for XOR-splits by error

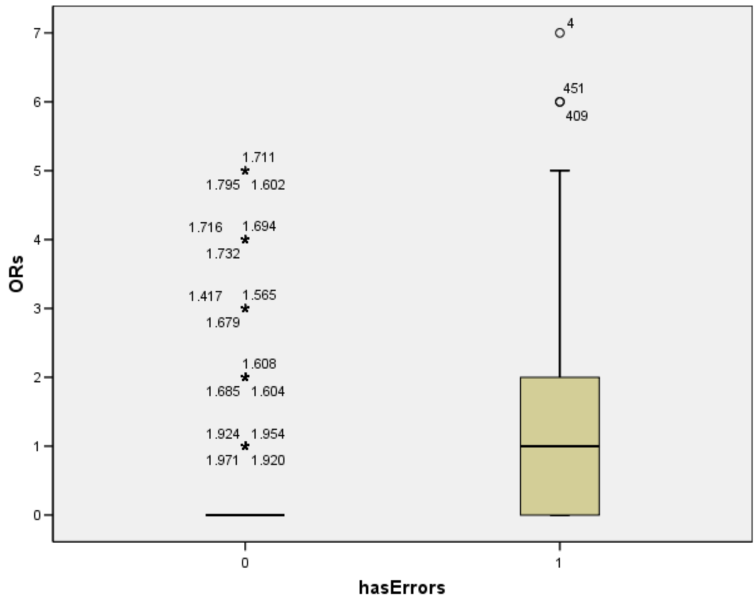


Figure C.62: Box plot for OR-splits by error

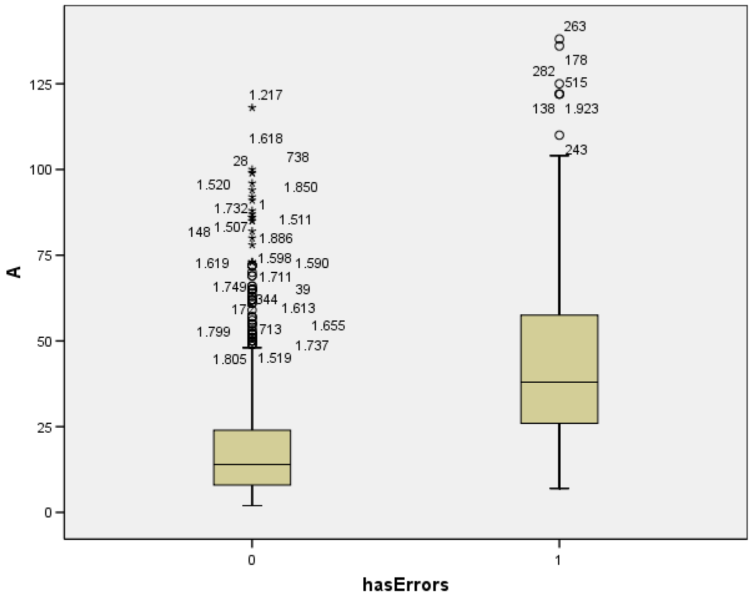


Figure C.63: Box plot for arcs A by error

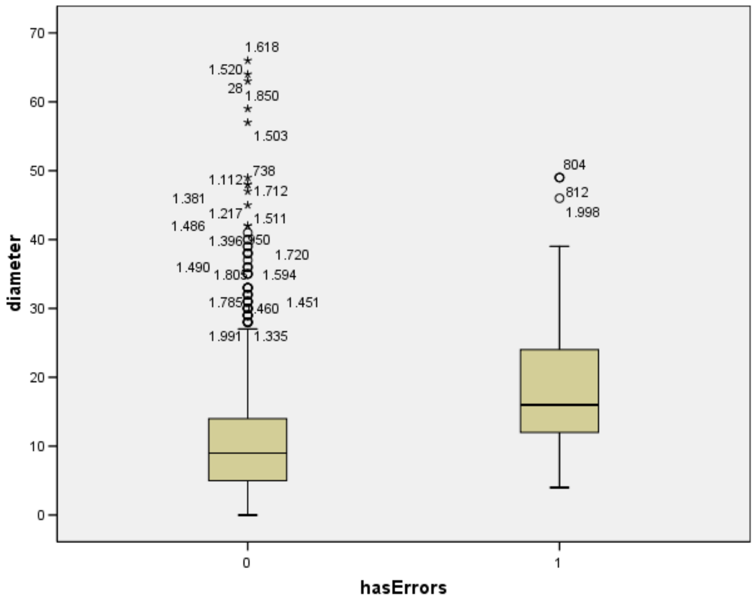


Figure C.64: Box plot for diameter by error

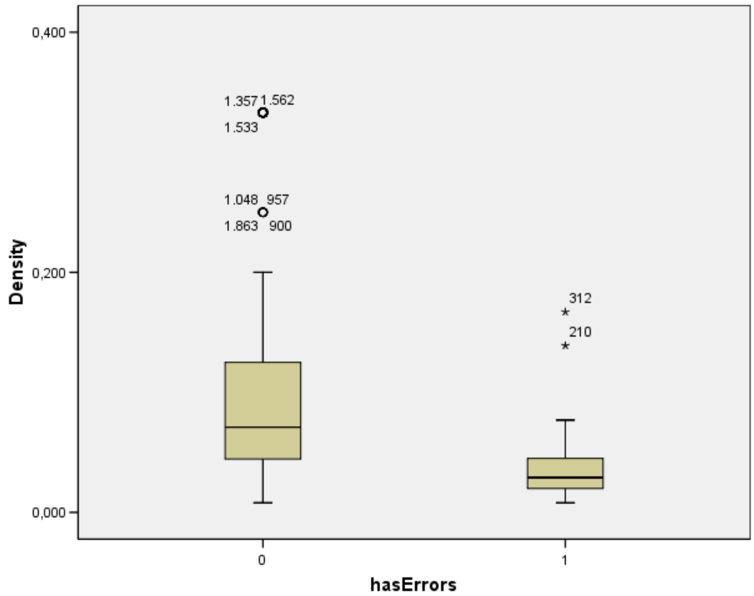


Figure C.65: Box plot for density by error

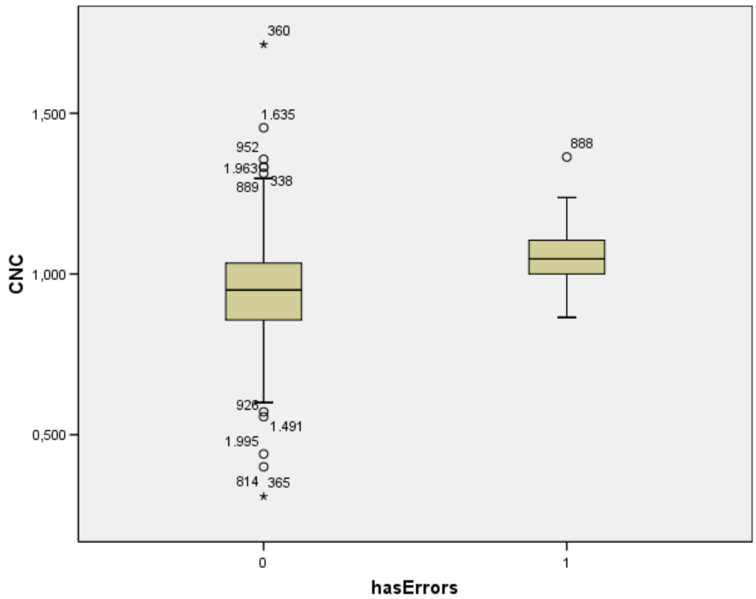


Figure C.66: Box plot for coefficient of connectivity CNC by error

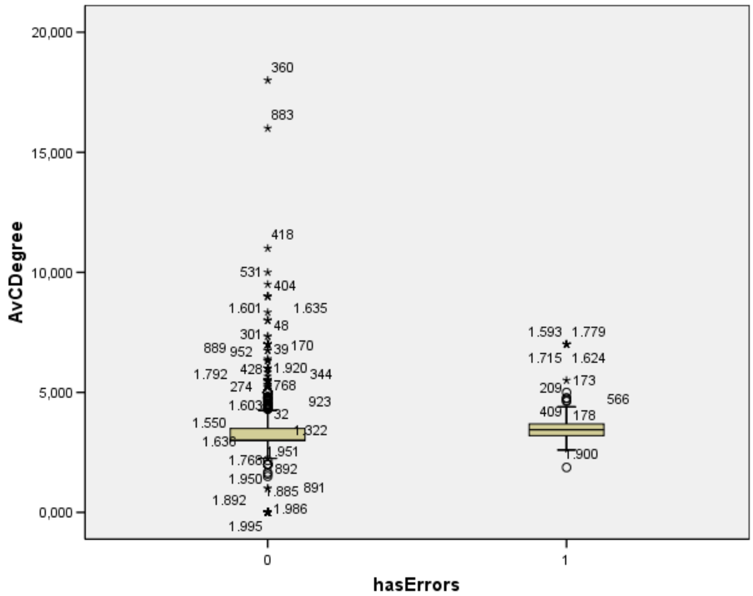


Figure C.67: Box plot for average connector degree by error

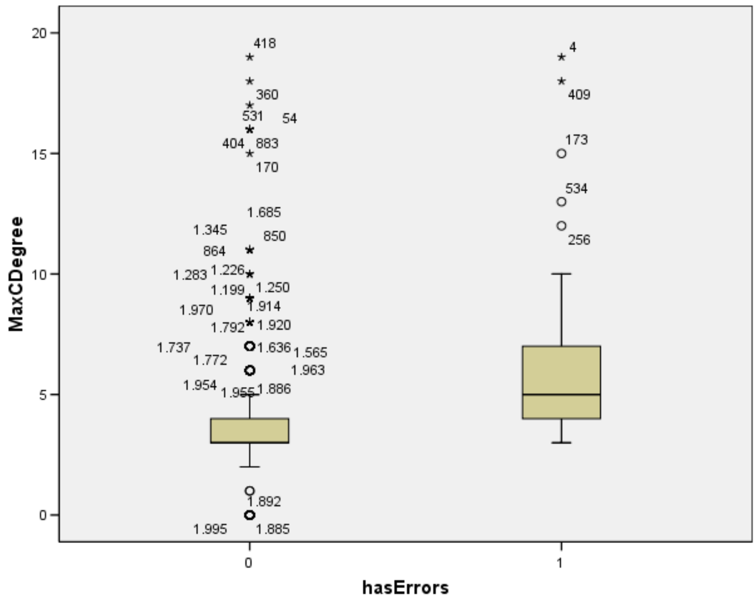


Figure C.68: Box plot for maximum connector degree by error

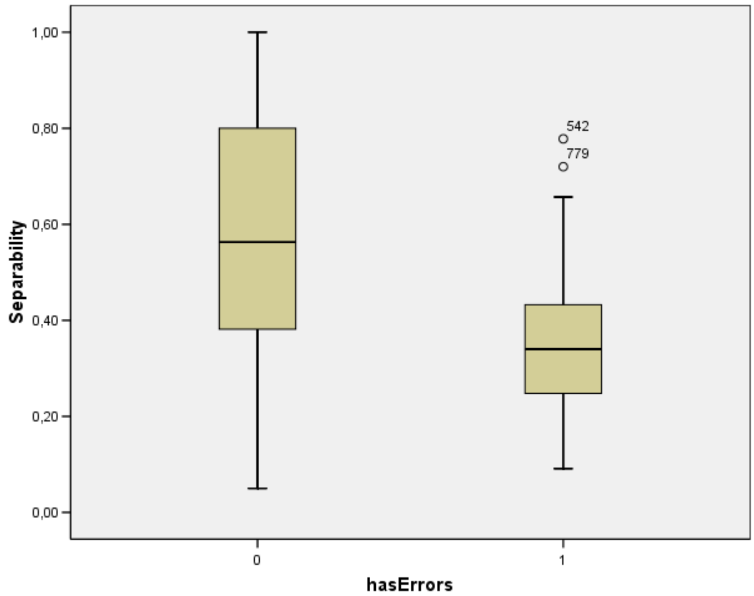


Figure C.69: Box plot for separability by error

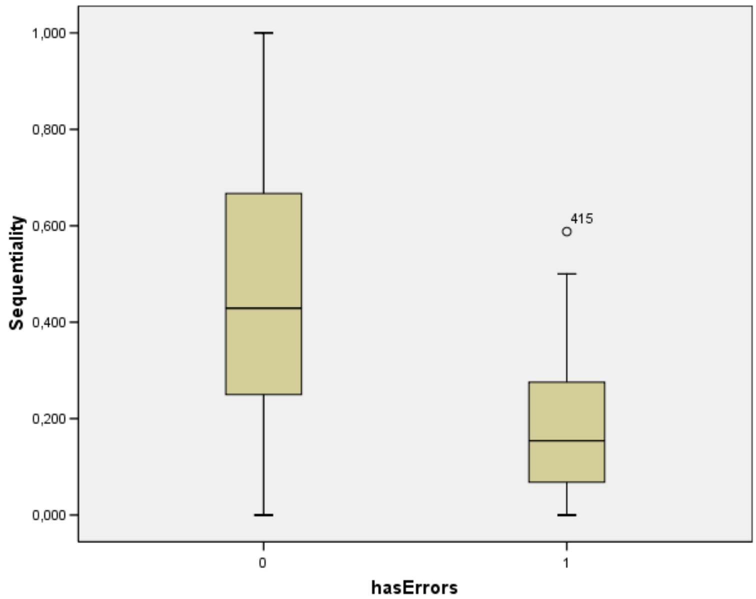


Figure C.70: Box plot for sequentiality by error

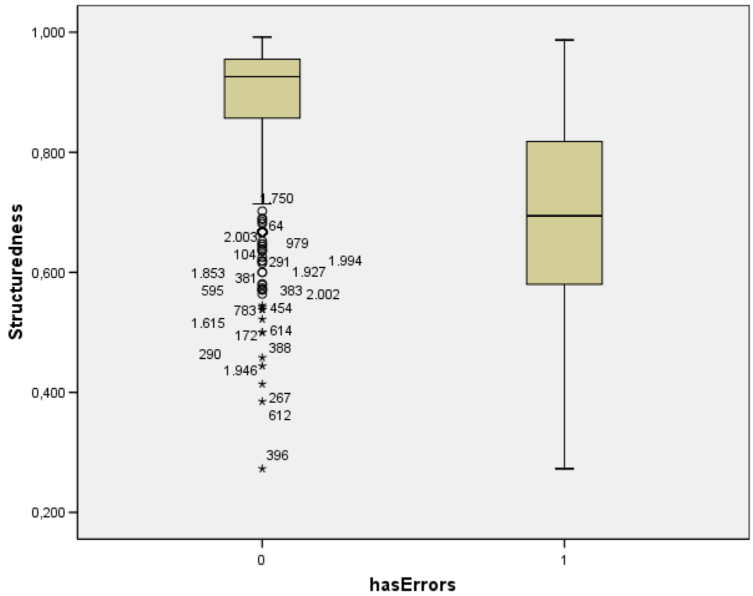


Figure C.71: Box plot for structuredness by error

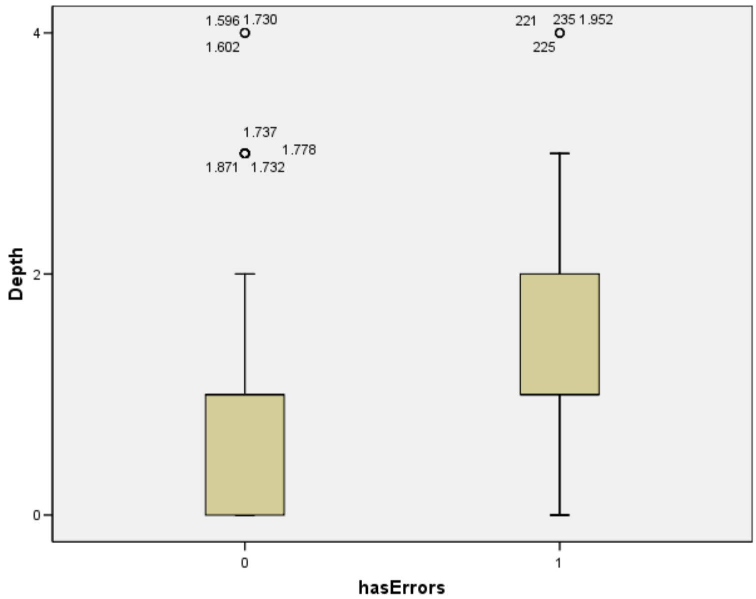


Figure C.72: Box plot for depth by error

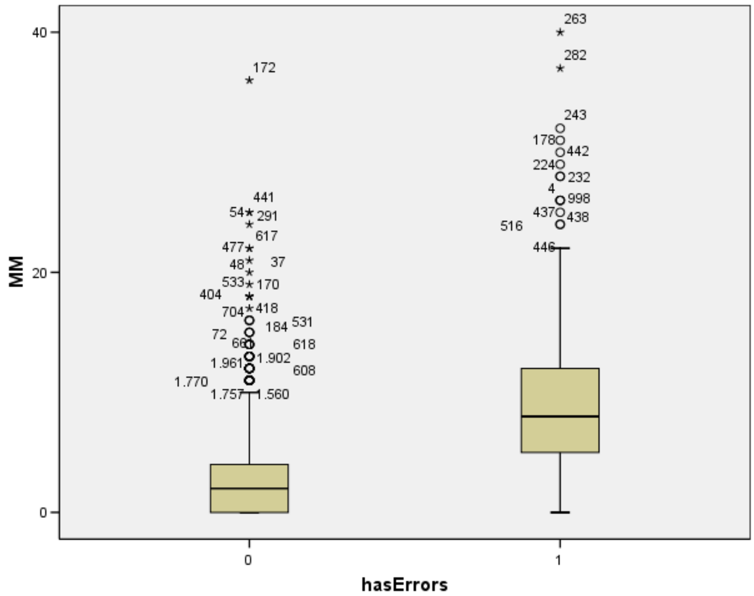


Figure C.73: Box plot for connector mismatch MM by error

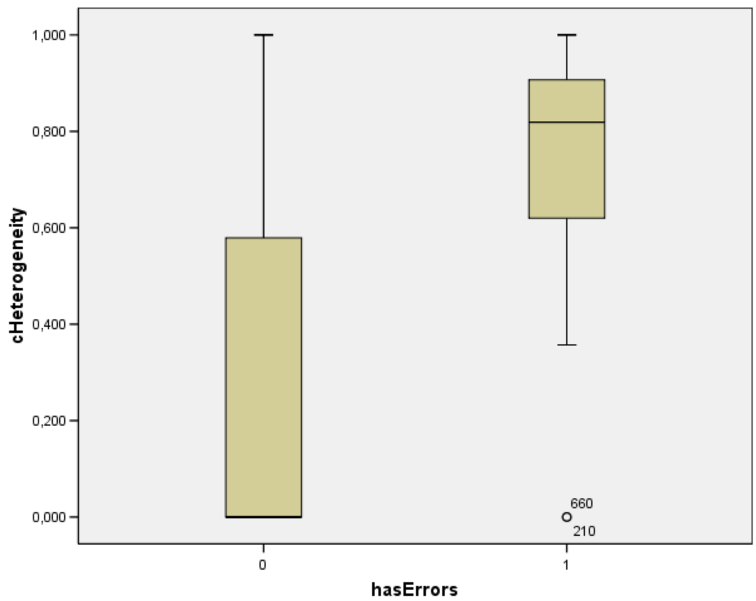


Figure C.74: Box plot for connector heterogeneity by error

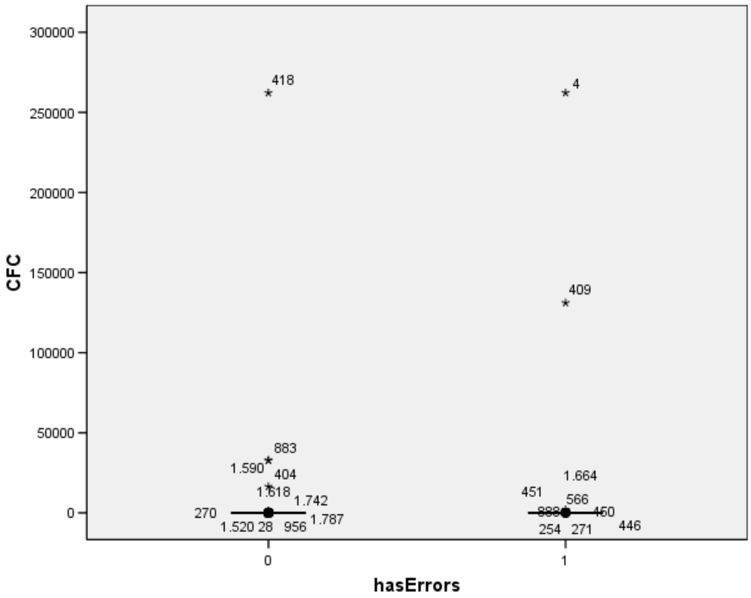


Figure C.75: Box plot for control flow complexity CFC by error

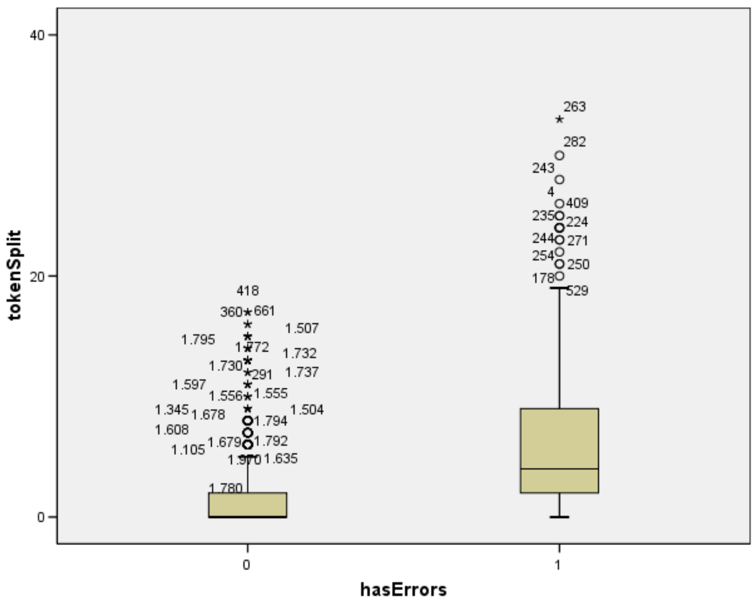


Figure C.76: Box plot for token split by error

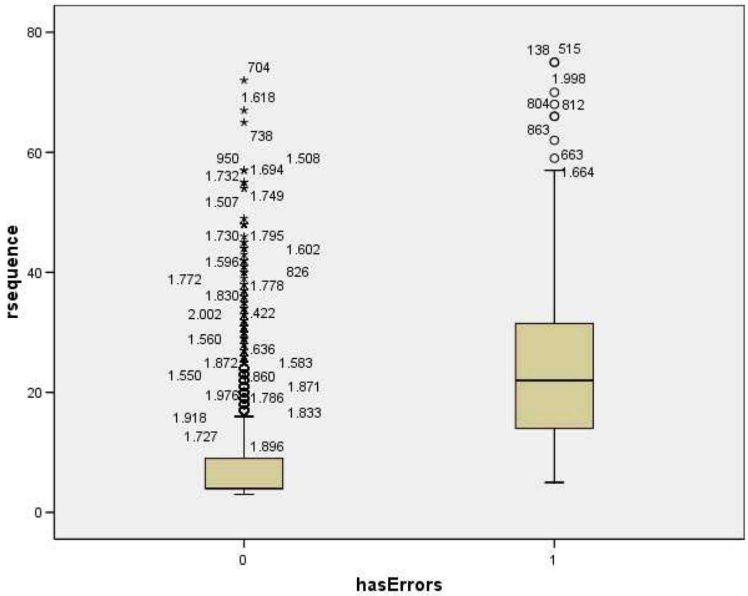


Figure C.77: Box plot for trivial construct rule application by error

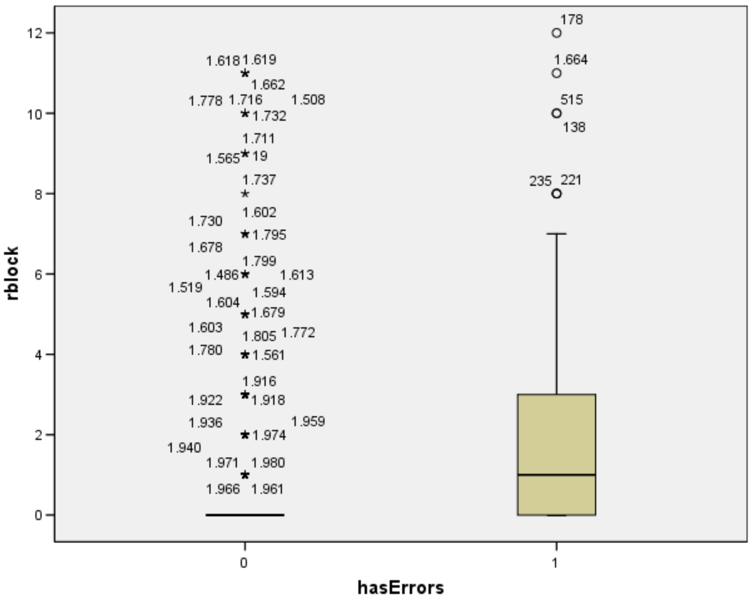


Figure C.78: Box plot for structured block rule application by error

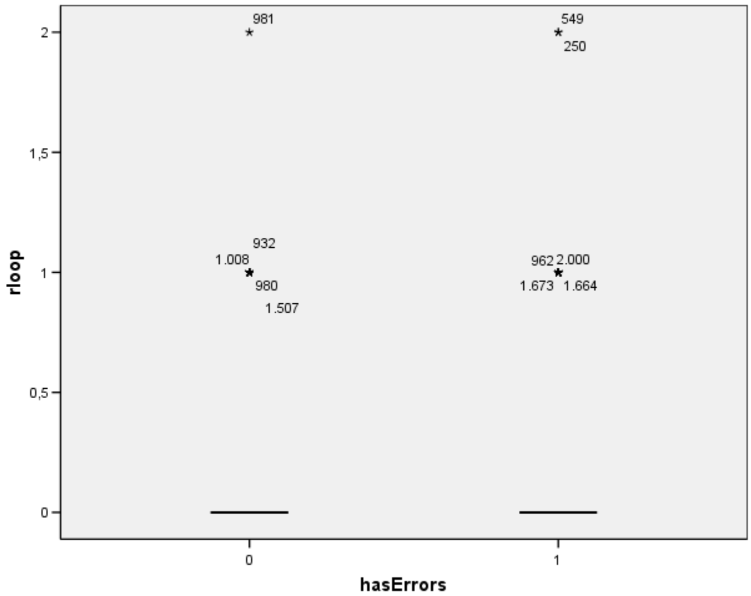


Figure C.79: Box plot for structured loop rule application by error

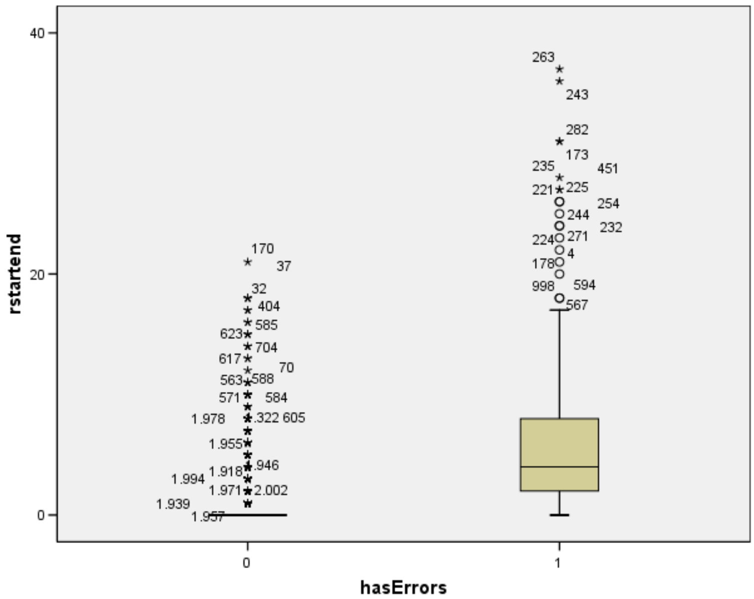


Figure C.80: Box plot for structured start and end rule application by error

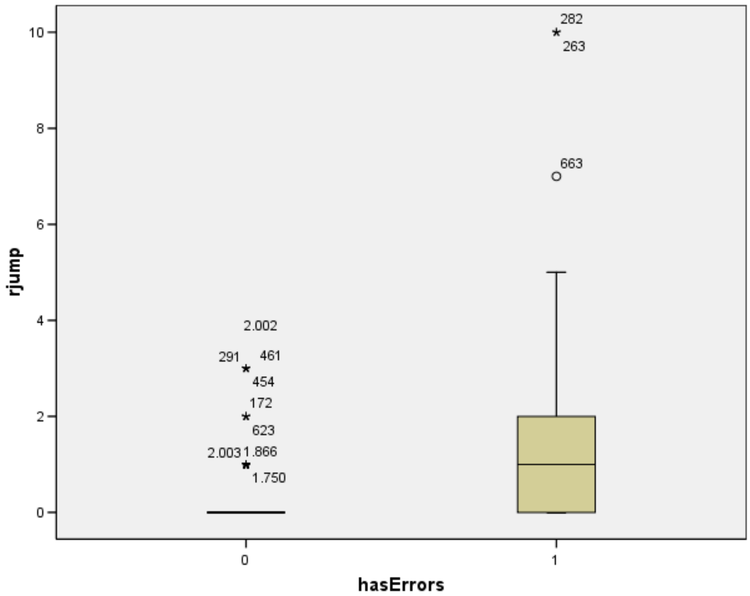


Figure C.81: Box plot for unstructured start and end rule application by error

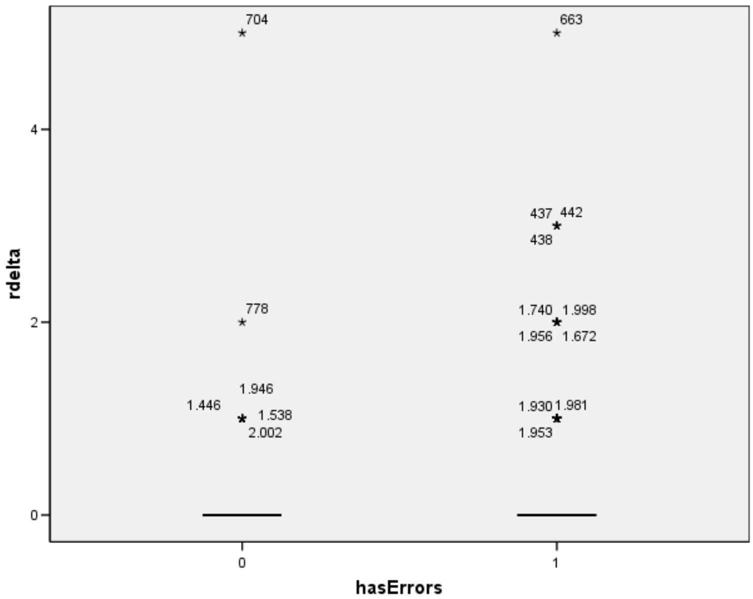


Figure C.82: Box plot for delta rule application by error

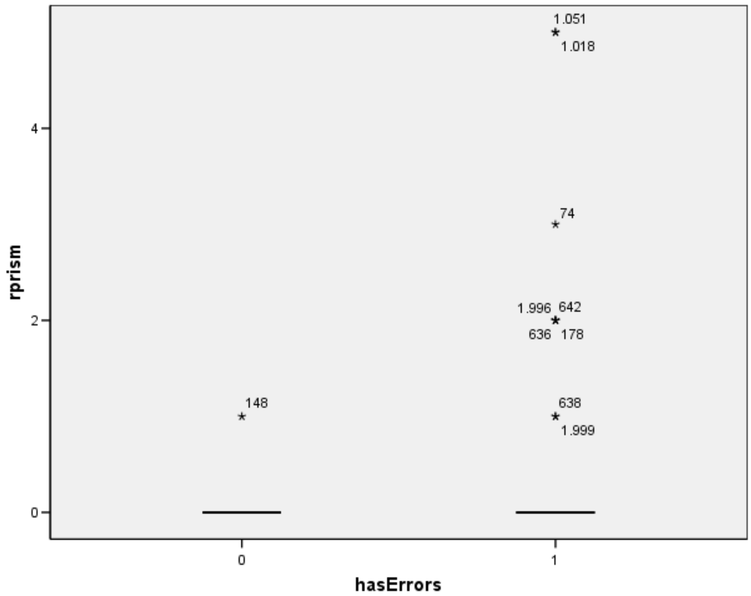


Figure C.83: Box plot for prism rule application by error

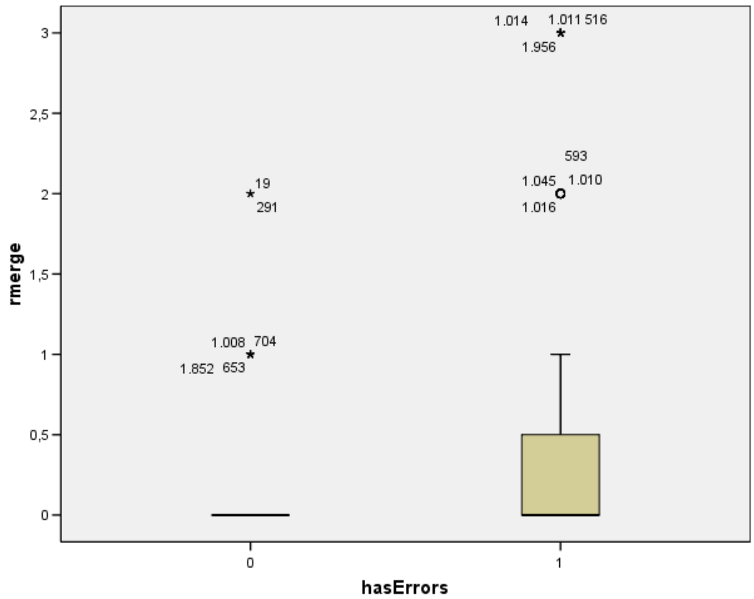


Figure C.84: Box plot for connector merge rule application by error

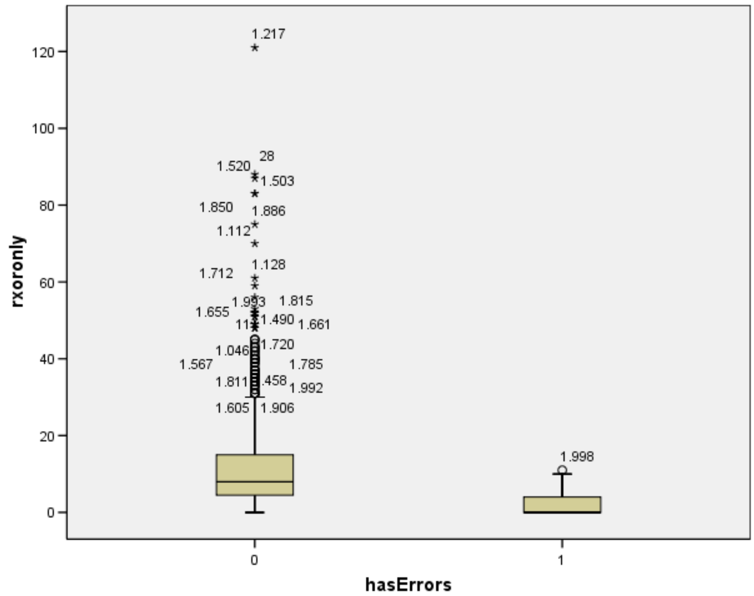


Figure C.85: Box plot for homogeneous rule application by error

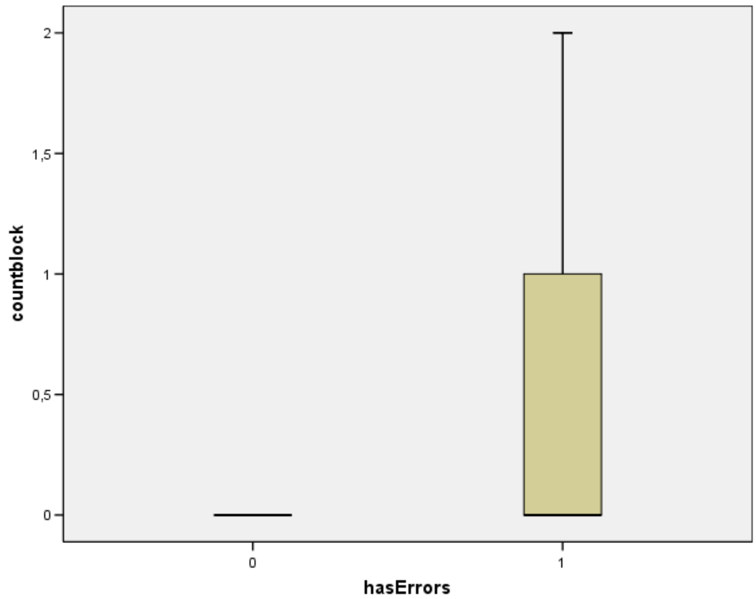


Figure C.86: Box plot for structured block errors by error

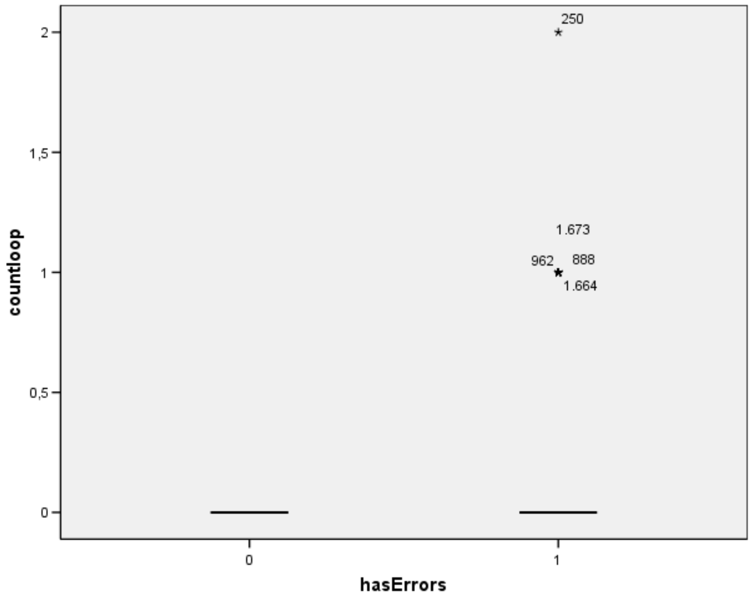


Figure C.87: Box plot for structured loop errors by error

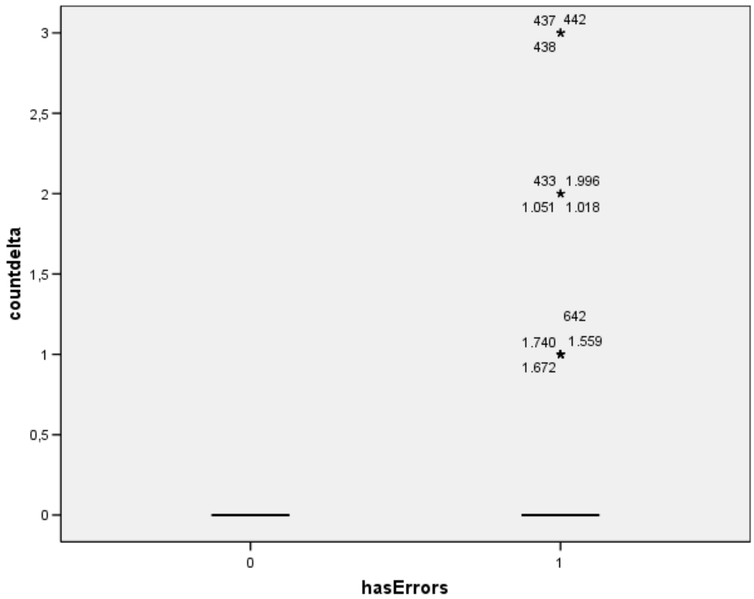


Figure C.88: Box plot for delta errors by error

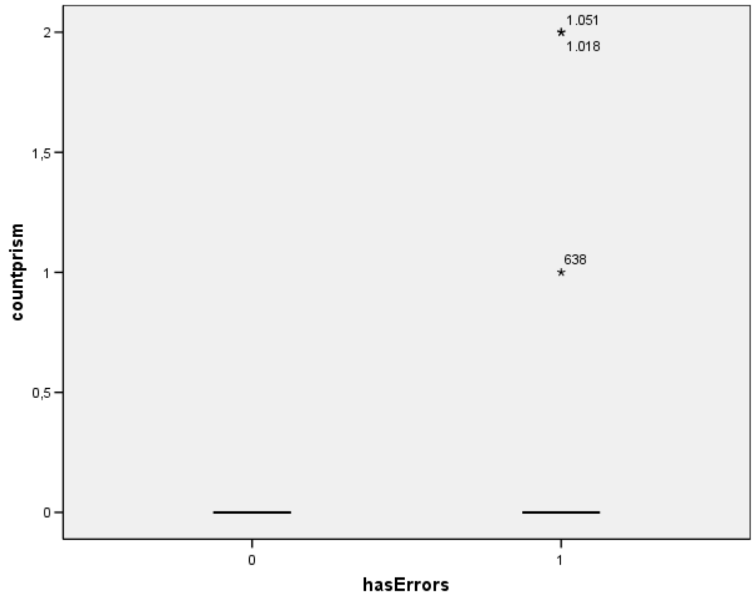


Figure C.89: Box plot for prism errors by error

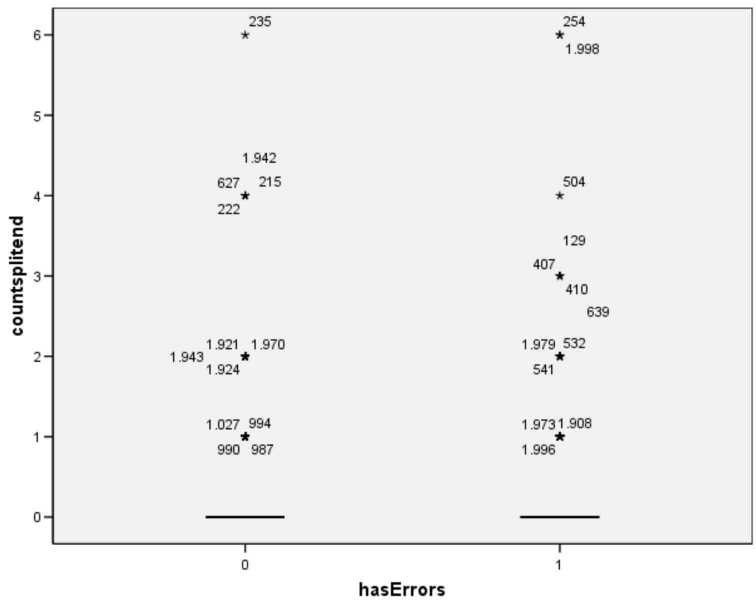


Figure C.90: Box plot for unstructured start and end errors by error

Table C.3: Results of Kolmogorov-Smironov test

	Mean	Std. Dev.	Z	Sig.		Mean	Std. Dev.	Z	Sig.
N	20,71	16,84	6,55	0,00	A	21,11	18,87	6,96	0,00
C	4,27	5,01	9,11	0,00	Sequentiality	0,46	0,31	6,04	0,00
E	10,47	8,66	7,35	0,00	CNC	0,96	0,13	4,91	0,00
Es	2,43	2,70	13,08	0,00	Density	0,09	0,07	7,00	0,00
Ee	2,77	3,20	12,80	0,00	tokenSplit	1,82	3,53	13,57	0,00
F	5,98	4,94	7,29	0,00	AvCDegree	2,88	1,60	14,49	0,00
AND	1,26	2,24	12,81	0,00	MaxCDegree	3,56	2,40	10,34	0,00
XOR	2,25	3,00	10,15	0,00	MM	3,31	4,55	10,45	0,00
OR	0,76	1,54	15,79	0,00	CYC	0,01	0,08	23,59	0,00
ANDj	0,63	1,23	16,28	0,00	Separability	0,56	0,27	4,73	0,00
XORj	1,01	1,46	11,54	0,00	Depth	0,70	0,74	12,05	0,00
ORj	0,37	0,82	18,98	0,00	Structuredness	0,88	0,11	9,01	0,00
ANDs	0,62	1,17	16,14	0,00	CFC	382,62	8849,48	22,11	0,00
XORs	1,24	1,75	11,54	0,00	cHeterogeneity	0,28	0,35	16,66	0,00
ORs	0,37	0,86	19,32	0,00	diameter	11,45	8,21	5,98	0,00

C.4 Analysis of Variance for Metrics grouped by hasErrors

This section summarizes the result of the analysis of variance for metrics grouped by hasErrors. First, we conduct the Kolmogorov-Smirnov test to verify that all variables follow a normal distribution. Then, we summarize the results of the analysis of variance showing that the mean values are significantly different for all metrics.

Table C.4: Analysis of Variance Results ordered by F-Statistic Values

	F	Sig.		F	Sig.
C	884,41	0,00	Depth	286,19	0,00
ANDj	824,72	0,00	ORs	264,28	0,00
AND	819,96	0,00	XORs	232,24	0,00
Structuredness	780,13	0,00	Sequentiality	223,45	0,00
MM	627,43	0,00	MaxCDegree	198,20	0,00
cHeterogeneity	585,51	0,00	diameter	180,17	0,00
E	563,04	0,00	OR	176,35	0,00
Ee	540,36	0,00	Separability	172,92	0,00
N	532,05	0,00	Density	156,89	0,00
A	518,24	0,00	CNC	137,23	0,00
ANDs	502,87	0,00	CYC	124,69	0,00
tokenSplit	471,12	0,00	F	66,59	0,00
Es	424,41	0,00	ORj	64,25	0,00
XORj	344,48	0,00	AvCDegree	44,86	0,00
XOR	331,22	0,00	CFC	6,95	0,01

C.5 Correlation between hasErrors and Metrics

This section shows the correlation between hasErrors and the different metrics, first as Pearson's correlation coefficient (Table C.5) and afterwards as Spearman's rank correlation coefficient (Table C.6).

Table C.5: Pearson Correlation between hasErrors and Metrics (below significance)

	hasErrors		hasErrors
Duration	0,13	ORs	0,34
	0,00		0,00
Restsize	0,62	A	0,45
	0,00		0,00
N	0,46	diameter	0,29
	0,00		0,00
C	0,55	Density	-0,27
	0,00		0,00
E	0,47	CNC	0,25
	0,00		0,00
Es	0,42	AvCDegree	0,15
	0,00		0,00
Ee	0,46	MaxCDegree	0,30
	0,00		0,00
F	0,18	Separability	-0,28
	0,00		0,00
AND	0,54	Sequentiality	-0,32
	0,00		0,00
XOR	0,38	Structuredness	-0,53
	0,00		0,00
OR	0,28	Depth	0,35
	0,00		0,00
ANDj	0,54	MM	0,49
	0,00		0,00
XORj	0,38	cHeterogeneity	0,48
	0,00		0,00
ORj	0,18	CFC	0,06
	0,00		0,01
ANDs	0,45	CYC	0,24
	0,00		0,00
XORs	0,32	tokenSplit	0,44
	0,00		0,00

Table C.6: Spearman Rank Correlation between hasErrors and Metrics (below significance)

	hasErrors		hasErrors
Duration	0,19 0,00	ORs	0,31 0,00
Restsize	0,66 0,00	A	0,38 0,00
N	0,38 0,00	diameter	0,30 0,00
C	0,43 0,00	Density	-0,37 0,00
E	0,38 0,00	CNC	0,28 0,00
Es	0,35 0,00	AvCDegree	0,23 0,00
Ee	0,38 0,00	MaxCDegree	0,33 0,00
F	0,19 0,00	Separability	-0,29 0,00
AND	0,45 0,00	Sequentiality	-0,35 0,00
XOR	0,35 0,00	Structuredness	-0,36 0,00
OR	0,30 0,00	Depth	0,34 0,00
ANDj	0,48 0,00	MM	0,42 0,00
XORj	0,33 0,00	cHeterogeneity	0,46 0,00
ORj	0,15 0,00	CFC	0,39 0,00
ANDs	0,37 0,00	CYC	0,30 0,00
XORs	0,31 0,00	tokenSplit	0,38 0,00

Appendix D

Logistic Regression Results

This appendix gathers details of the logistic regression analysis. In particular, Section D.1 gives a tabular overview of the collinearity analysis of the variables. This analysis led to a reduction of the variable set in such a way that S_N is the only remaining count metric for size. Section C.2 presents the results of univariate logistic regression models of all variables of the reduced set. These univariate models show that there is no constant in a multivariate model required since the constant is not significantly different from zero in two models (see Wald statistic). Furthermore, the control flow complexity is not significantly different from zero in both models with and without constant. Therefore, it is dropped from the variables list. Section D.3 shows results from the multivariate logistic regression analysis.

D.1 Collinearity Analysis

This section gives the results of the collinearity analysis. The absence of collinearity is not a hard criterion for the applicability of logistic regression, but it is desirable. In a

Table D.1: Tolerance Values for Metrics

	Tolerance		Tolerance
N	0.0000	A	0.0017
C	0.0000	diameter	0.1217
E	0.0062	Density	0.1978
Es	0.1269	CNC	0.1362
Ee	0.0607	AvCDegree	0.1151
F	0.0228	MaxCDegree	0.0792
AND	0.0064	Separability	0.2539
XOR	0.0123	Sequentiality	0.1377
OR	0.0125	Structuredness	0.5555
ANDj	0.0202	Depth	0.2228
XORj	0.0431	MM	0.2365
ORj	0.0404	cHeterogeneity	0.3824
ANDs	0.0209	CFC	0.6966
XORs	0.0287	CYC	0.8913
ORs	0.0349	tokenSplit	0.0488

Table D.2: Tolerance Values after reducing the Metrics Set

	Tolerance		Tolerance
N	0.0931	Structuredness	0.6225
diameter	0.1564	Depth	0.2606
CNC	0.2570	MM	0.3261
Density	0.2875	cHeterogeneity	0.4241
AvCDegree	0.1283	CFC	0.8073
MaxCDegree	0.1080	CYC	0.9326
Separability	0.2828	tokenSplit	0.3008
Sequentiality	0.2576		

variable set without collinearity every variable should have a tolerance value higher than 0.1, otherwise there is a collinearity problem. In the original variable set (Table D.1) there are several collinearity problems. We dropped the count metrics apart from S_N since they were highly correlated. This resulted in a reduced variable set with almost no collinearity problems (Table D.2). The S_N metric is close to the 0.1 threshold and therefore kept in the metrics set.

D.2 Univariate Logistic Regression

This section presents the results of the univariate logistic regression analysis. In particular we calculated univariate models with and without a constant (see Tables D.3 and D.4). As a conclusion from these models we drop the constant and the control flow complexity CFC for the multivariate analysis. First, the constant is not significantly different from zero (see Wald statistic) in the separability and the sequentiality model which suggests that it is not necessary. Second, the CFC metric is not significantly different from zero (see Wald statistic) in both models with and without constant.

D.3 Multivariate Logistic Regression

Based on a reduced set of variables without CFC we calculated multivariate logistic regression models. Figure D.1 shows that the Hosmer & Lemeshow Test indicates a good fit based on the difference between observed and predicted frequencies. This test should yield a value greater than 5% and this condition is fulfilled by all models from step 3 on. Figure D.2 summarizes the value of Nagelkerke's R^2, a statistic ranging from 0 to 1 that serves as a coefficient of determination. It indicates which fraction of the variability is explained. The figure shows that from step 3 on the value approaches 0.90 which is an excellent value. Figure D.3 and D.4 give the classification tables and the equations of the models in the different steps.

Table D.3: Univariate logistic regression models without constant

	B	Exp(B)	Wald	Hosmer & L.	Nagelkerke R^2
N	-0.440	0.957	0.000	0.000	0.256
diameter	-0.112	0.894	0.000	0.000	0.387
CNC	-2.082	0.013	0.000	0.000	0.637
Density	-41.081	0.000	0.000	0.000	0.771
AvCDegree	-0.532	0.588	0.000	0.000	0.506
MaxCDegree	-0.351	0.704	0.000	0.000	0.396
Separability	-4.657	0.009	0.000	0.000	0.733
Sequentiality	-7.038	0.001	0.000	0.123	0.760
Structuredness	-2.688	0.068	0.000	0.000	0.728
Depth	-0.908	0.403	0.000	0.000	0.193
MM	-0.090	0.914	0.000	0.000	0.066
cHeterogeneity	-1.223	0.294	0.000	0.000	0.085
CFC	0.000	1.000	0.531	0.000	0.000
CYC	0.301	1.352	0.588	0.999	0.000
tokenSplit	-0.067	0.935	0.000	0.000	0.020

Table D.4: Univariate logistic regression models with constant

	Cons.	Exp(Cons.)	Wald	B	Exp(B)	Wald	H. & L.	N. R^2
N	-3.954	0.019	0.000	0.068	1.070	0.000	0.000	0.295
diameter	-3.306	0.037	0.000	0.087	1.091	0.000	0.000	0.132
CNC	-9.411	0.000	0.000	7.294	1472.146	0.000	0.000	0.138
Density	0.634	1.885	0.001	-54.440	0.000	0.000	0.000	0.311
AvCDegree	-3.029	0.048	0.000	0.291	1.338	0.000	0.000	0.042
MaxCDegree	3.575	0.028	0.000	0.344	1.411	0.000	0.000	0.145
Separability	0.027	1.028	0.872	-4.716	0.009	0.000	0.000	0.184
Sequentiality	-0.204	0.815	0.117	-6.391	0.002	0.000	0.262	0.268
Structuredness	7.064	1169.081	0.000	-11.210	0.000	0.000	0.000	0.377
Depth	-3.419	0.033	0.000	1.343	3.830	0.000	0.000	0.208
MM	-3.459	0.031	0.000	0.270	1.310	0.000	0.000	0.318
cHeterogeneity	-4.811	0.008	0.000	5.259	192.361	0.000	0.000	0.413
CFC	-2.115	0.121	0.000	0.000	1.000	0.382	0.000	0.001
CYC	-2.244	0.106	0.000	5.104	164.740	0.000	0.999	0.065
tokenSplit	-2.871	0.057	0.000	0.269	1.308	0.000	0.000	0.235

Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	330,522	8	,000
2	26,819	8	,001
3	4,278	8	,831
4	4,341	8	,825
5	8,101	8	,424
6	9,961	8	,268
7	7,184	8	,517
8	10,573	8	,227
9	7,890	8	,444

Figure D.1: Hosmer and Lemeshow test for multivariate logistic regression

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	1178,396[a]	,546	,728
2	768,884[b]	,631	,841
3	584,495[c]	,664	,885
4	554,211[d]	,669	,892
5	528,702[c]	,673	,898
6	521,807[c]	,674	,899
7	515,520[d]	,675	,901
8	511,687[d]	,676	,901
9	513,645[d]	,676	,901

a. Estimation terminated at iteration number 5 because parameter estimates changed by less than ,001.

b. Estimation terminated at iteration number 6 because parameter estimates changed by less than ,001.

c. Estimation terminated at iteration number 7 because parameter estimates changed by less than ,001.

d. Estimation terminated at iteration number 8 because parameter estimates changed by less than ,001.

Figure D.2: Nagelkerke R^2 for multivariate logistic regression

Classification Table[a]

Observed			Predicted: hasErrors 0	Predicted: hasErrors 1	Percentage Correct
Step 1	hasErrors	0	1761	0	100,0
		1	213	0	,0
	Overall Percentage				89,2
Step 2	hasErrors	0	1736	25	98,6
		1	134	79	37,1
	Overall Percentage				91,9
Step 3	hasErrors	0	1720	41	97,7
		1	83	130	61,0
	Overall Percentage				93,7
Step 4	hasErrors	0	1719	42	97,6
		1	77	136	63,8
	Overall Percentage				94,0
Step 5	hasErrors	0	1719	42	97,6
		1	64	149	70,0
	Overall Percentage				94,6
Step 6	hasErrors	0	1721	40	97,7
		1	61	152	71,4
	Overall Percentage				94,9
Step 7	hasErrors	0	1722	39	97,8
		1	61	152	71,4
	Overall Percentage				94,9
Step 8	hasErrors	0	1723	38	97,8
		1	57	156	73,2
	Overall Percentage				95,2
Step 9	hasErrors	0	1724	37	97,9
		1	58	155	72,8
	Overall Percentage				95,2

a. The cut value is ,500

Figure D.3: Classification table for multivariate logistic regression

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	Structuredness	-2,688	,093	843,418	1	,000	,068
Step 2[b]	N	,084	,005	247,455	1	,000	1,088
	Structuredness	-5,466	,237	530,121	1	,000	,004
Step 3[c]	N	,053	,006	73,718	1	,000	1,054
	Structuredness	-7,270	,387	353,553	1	,000	,001
	cHeterogeneity	4,419	,398	123,375	1	,000	83,029
Step 4[d]	N	,054	,006	73,082	1	,000	1,056
	CYC	4,392	,831	27,915	1	,000	80,835
	Structuredness	-7,495	,409	335,352	1	,000	,001
	cHeterogeneity	4,364	,411	112,589	1	,000	78,600
Step 5[e]	N	,043	,007	40,881	1	,000	1,044
	CNC	3,404	,712	22,878	1	,000	30,070
	CYC	3,995	,862	21,484	1	,000	54,342
	Structuredness	-10,333	,748	190,748	1	,000	,000
	cHeterogeneity	3,244	,457	50,273	1	,000	25,629
Step 6[f]	N	,039	,007	31,900	1	,000	1,040
	CNC	3,320	,708	22,013	1	,000	27,654
	MM	,067	,026	6,560	1	,010	1,069
	CYC	4,264	,873	23,857	1	,000	71,071
	Structuredness	-10,217	,744	188,622	1	,000	,000
	cHeterogeneity	2,778	,491	32,029	1	,000	16,084
Step 7[g]	N	,033	,007	21,363	1	,000	1,034
	CNC	3,898	,738	27,906	1	,000	49,285
	MM	,069	,025	7,407	1	,006	1,072
	CYC	3,825	,890	18,466	1	,000	45,852
	Separability	-1,648	,670	6,059	1	,014	,192
	Structuredness	-9,869	,757	169,882	1	,000	,000
	cHeterogeneity	2,723	,490	30,895	1	,000	15,222
Step 8[h]	N	,016	,011	1,946	1	,163	1,016
	CNC	3,805	,753	25,543	1	,000	44,919
	MM	,081	,026	9,670	1	,002	1,085
	CYC	3,601	,900	16,028	1	,000	36,642
	Separability	-1,980	,712	7,738	1	,005	,138
	Structuredness	-9,893	,760	169,376	1	,000	,000
	cHeterogeneity	2,882	,505	32,605	1	,000	17,849
	diameter	,041	,021	3,867	1	,049	1,042
Step 9[h]	CNC	4,008	,742	29,193	1	,000	55,033
	MM	,094	,025	14,572	1	,000	1,098
	CYC	3,409	,891	14,648	1	,000	30,248
	Separability	-2,338	,673	12,058	1	,001	,096
	Structuredness	-9,957	,760	171,551	1	,000	,000
	cHeterogeneity	3,003	,501	35,988	1	,000	20,139
	diameter	,064	,013	24,474	1	,000	1,066

a. Variable(s) entered on step 1: Structuredness.

b. Variable(s) entered on step 2: N.

c. Variable(s) entered on step 3: cHeterogeneity.

d. Variable(s) entered on step 4: CYC.

e. Variable(s) entered on step 5: CNC.

f. Variable(s) entered on step 6: MM.

g. Variable(s) entered on step 7: Separability.

h. Variable(s) entered on step 8: diameter.

Figure D.4: Equation of multivariate logistic regression models

D.4 Second Best Logistic Regression

After excluding the metrics of the regression model of Section D.3, i.e. without the coefficient of network connectivity CNC, connector mismatch MM, cyclicity CYC, separability Π, structuredness Φ, connector heterogeneity CH, and without the diameter $diam$, we calculated a second best multivariate logistic regression models. This model includes sequentiality Ξ, density Δ, and size S_N. Figure D.5 shows that the Hosmer & Lemeshow Test fails to indicate a good fit since the value is less than 5% after the second model. Figure D.6 summarizes the value of Nagelkerke's R^2 that indicates still a high fraction of explanation of the variability with a value of 0.824. Figure D.7 and D.8 give the classification tables and the equations of the models in the different steps.

Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	11,389	7	,123
2	92,939	8	,000
3	18,614	8	,017

Figure D.5: Hosmer and Lemeshow test for second best multivariate logistic regression

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	1071,748[a]	,570	,760
2	945,296[a]	,596	,795
3	835,472[b]	,618	,824

a. Estimation terminated at iteration number 7 because parameter estimates changed by less than ,001.

b. Estimation terminated at iteration number 8 because parameter estimates changed by less than ,001.

Figure D.6: Nagelkerke R^2 for second best multivariate logistic regression

Classification Table[a]

Observed			Predicted hasErrors 0	Predicted hasErrors 1	Percentage Correct
Step 1	hasErrors	0	1703	58	96,7
		1	204	9	4,2
	Overall Percentage				86,7
Step 2	hasErrors	0	1761	0	100,0
		1	213	0	,0
	Overall Percentage				89,2
Step 3	hasErrors	0	1725	36	98,0
		1	134	79	37,1
	Overall Percentage				91,4

a. The cut value is ,500

Figure D.7: Classification table for second best multivariate logistic regression

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	Sequentiality	-7,038	,315	498,244	1	,000	,001
Step 2[b]	Sequentiality	-3,596	,413	75,916	1	,000	,027
	Density	-20,822	2,327	80,046	1	,000	,000
Step 3[c]	Sequentiality	-6,540	,594	121,362	1	,000	,001
	Density	-23,873	2,590	84,992	1	,000	,000
	N	,034	,004	87,631	1	,000	1,034

a. Variable(s) entered on step 1: Sequentiality.

b. Variable(s) entered on step 2: Density.

c. Variable(s) entered on step 3: N.

Figure D.8: Equation of multivariate second best logistic regression models

D.5 Third Best Logistic Regression

After excluding the metrics of the regression model of Sections D.3 and D.4, i.e. only with token split TS, average and maximum connector degree $\overline{d_C}$ and $\widehat{d_C}$, and Depth Λ, we calculated a third best multivariate logistic regression models. Figure D.9 shows that the Hosmer & Lemeshow Test fails to indicate a good fit since the value is less than 5% after the second model. Figure D.10 summarizes the value of Nagelkerke's R^2 that indicates still a high fraction of explanation of the variability with a value of 0.627. Figure D.11 and D.12 give the classification tables and the equations of the models in the different steps.

Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	528,875	6	,000
2	389,011	7	,000
3	376,036	7	,000
4	363,645	7	,000

Figure D.9: Hosmer and Lemeshow test for third best multivariate logistic regression

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	1793,593[a]	,380	,506
2	1529,029[b]	,458	,610
3	1496,768[b]	,466	,622
4	1481,988[b]	,470	,627

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

b. Estimation terminated at iteration number 5 because parameter estimates changed by less than ,001.

Figure D.10: Nagelkerke R^2 for third best multivariate logistic regression

Classification Table[a]

	Observed		Predicted: hasErrors 0	Predicted: hasErrors 1	Predicted: Percentage Correct
Step 1	hasErrors	0	1414	347	80,3
		1	213	0	,0
	Overall Percentage				71,6
Step 2	hasErrors	0	1390	371	78,9
		1	164	49	23,0
	Overall Percentage				72,9
Step 3	hasErrors	0	1385	376	78,6
		1	164	49	23,0
	Overall Percentage				72,6
Step 4	hasErrors	0	1385	376	78,6
		1	159	54	25,4
	Overall Percentage				72,9

a. The cut value is ,500

Figure D.11: Classification table for third best multivariate logistic regression

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	AvCDegree	-,532	,021	615,699	1	,000	,588
Step 2[b]	tokenSplit	,319	,024	183,587	1	,000	1,376
	AvCDegree	-,814	,033	602,859	1	,000	,443
Step 3[c]	tokenSplit	,222	,028	64,643	1	,000	1,248
	AvCDegree	-1,294	,093	195,393	1	,000	,274
	MaxCDegree	,425	,073	33,848	1	,000	1,530
Step 4[d]	tokenSplit	,194	,029	44,607	1	,000	1,214
	AvCDegree	-1,371	,097	200,313	1	,000	,254
	MaxCDegree	,405	,074	29,983	1	,000	1,500
	Depth	,440	,115	14,562	1	,000	1,553

a. Variable(s) entered on step 1: AvCDegree.

b. Variable(s) entered on step 2: tokenSplit.

c. Variable(s) entered on step 3: MaxCDegree.

d. Variable(s) entered on step 4: Depth.

Figure D.12: Equation of third best multivariate logistic regression models

Bibliography

[AAB+05] A. Arkin, S. Askary, B. Bloch, F. Curbera, Y. Goland, N. Kartha, C.K. Liu, S. Thatte, P. Yendluri, and A. Yiu. Web services business process execution language version 2.0. wsbpel-specification-draft-01, OASIS, September 2005.

[Aal97] W.M.P. van der Aalst. Verification of Workflow Nets. In Pierre Azéma and Gianfranco Balbo, editors, *Application and Theory of Petri Nets 1997*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426. Springer Verlag, 1997.

[Aal98] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[Aal99] W.M.P. van der Aalst. Formalization and Verification of Event-driven Process Chains. *Information and Software Technology*, 41(10):639–650, 1999.

[Aal03] W. M. P. van der Aalst. Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language. QUT Technical report FIT-TR-2003-06, Queensland University of Technology, Brisbane, 2003.

[ABCC05] W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors. *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume 3649, 2005.

[ACD+99] R. Anupindi, S. Chopra, S.D. Deshmukh, J.A. van Mieghem, and E. Zemel. *Managing Business Process Flows*. Prentice Hall, 1999.

[ACD+03] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems, 2003.

[ADH+03] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data & Knowledge Engineering*, 47(2):237–267, 2003.

[ADK02] W.M.P. van der Aalst, J. Desel, and E. Kindler. On the semantics of EPCs: A vicious circle. In M. Nüttgens and F.J. Rump, editor, *Proc. of the 1st GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2002), Trier, Germany*, pages 71–79, 2002.

[ADO00] W.M.P. van der Aalst, J. Desel, and A. Oberweis, editors. *Business Process Management, Models, Techniques, and Empirical Studies*, volume 1806 of *Lecture Notes in Computer Science*. Springer, 2000.

[AGRP07] E. Rolón Aguilar, F. García, F. Ruiz, and M. Piattini. An exploratory experiment to validate measures for business process models. In *First International Conference on Research Challenges in Information Science (RCIS)*, 2007.

[AH02] W.M.P. van der Aalst and K. van Hee. *Workflow Management: Models, Methods, and Systems*. The MIT Press, 2002.

[AH05] W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.

[AHKB03] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1):5–51, July 2003.

[AHV02] W.M.P. van der Aalst, A. Hirnschall, and H.M.W. Verbeek. An Alternative Way to Analyze Workflow Graphs. In A. Banks-Pidduck, J. Mylopoulos, C.C. Woo, and M.T. Ozsu, editors, *Proceedings of the 14th International Conference* *on Advanced Information Systems Engineering (CAiSE'02)*, volume 2348 of *LNCS*, pages 535–552. Springer-Verlag, Berlin, 2002.

[AHW03] W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske, editors. *Business Process Management, International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003, Proceedings*, volume 2678 of *Lecture Notes in Computer Science*. Springer, 2003.

[AJL05] W.M.P. van der Aalst, J.B. Jørgensen, and K.B. Lassen. Let's Go All the Way: From Requirements via Colored Workflow Nets to a BPEL Implementation of a New Bank System Paper. In R. Meersman and Z. Tari et al., editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005*, volume 3760 of *LNCS*, pages 22–39. Springer-Verlag, Berlin, 2005.

[AK01a] C. Atkinson and T. Kühne. The essence of multilevel metamodeling. In M. Gogolla and C. Kobryn, editors, *UML 2001 - The Unified Modeling Language, Modeling Languages, Concepts, and Tools, 4th International Conference, Toronto, Canada, October 1-5, 2001, Proceedings*, volume 2185 of *Lecture Notes in Computer Science*, pages 19–33. Springer, 2001.

[AK01b] C. Atkinson and T. Kühne. Processes and products in a multi-level metamodeling architecture. *International Journal of Software Engineering and Knowledge Engineering*, 11(6):761–783, 2001.

[AK03] C. Atkinson and T. Kühne. Model-driven development: A metamodeling foundation. *IEEE Software*, 20(5):36–41, 2003.

[AL05] W.M.P. van der Aalst and K.B. Lassen. Translating Workflow Nets to BPEL4WS. BETA Working Paper Series, WP 145, Eindhoven University of Technology, Eindhoven, 2005.

[Alb79] A.J. Albrecht. Measuring application development productivity. In *Proceeding IBM Applications Development Symposium, GUIDE Int and Share Inc., IBM Corp., Monterey, CA, Oct. 1417, 1979*, page 83, 1979.

[ARD+06] W.M.P. van der Aalst, V. Rubin, B.F. van Dongen, E. Kindler, and C.W. Günther. Process mining: A two-step approach using transition systems and regions. BPMCenter Report BPM-06-30, BPMcenter.org, 2006.

[ARGP06a] E. Rolón Aguilar, F. Ruiz, F. García, and M. Piattini. Applying software metrics to evaluate business process models. *CLEI Electron. J.*, 9, 2006.

[ARGP06b] E. Rolón Aguilar, F. Ruiz, F. García, and M. Piattini. Evaluation measures for business process models. In H. Haddad, editor, *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC), Dijon, France, April 23-27, 2006*, pages 1567–1568. ACM, 2006.

[ARGP06c] E. Rolón Aguilar, F. Ruiz, F. García, and M. Piattini. Towards a Suite of Metrics for Business Process Models in BPMN. In Y. Manolopoulos, J. Filipe, P. Constantopoulos, and J. Cordeiro, editors, *ICEIS 2006 - Proceedings of the Eighth International Conference on Enterprise Information Systems: Databases and Information Systems Integration (III), Paphos, Cyprus, May 23-27, 2006*, pages 440–443, 2006.

[AS02] A.D. Aczel and J. Sounderpandian. *Complete Business Statistics*. McGraw-Hill, 5th edition, 2002.

[AS07] T. Ami and R. Sommer. Comparison and evaluation of business process modelling and management tools. *International Journal of Services and Standards,*, 3(2):249261, 2007.

[ATM03] O. Adam, O. Thomas, and G. Martin. Fuzzy enhanced process management for the industrial order handling. In A.-W. Scheer, editor, *Proceedings of the 5th International Conference ; The Modern Information Technology in the Innovation Processes of the Industrial Enterprises, MITIP 2003*, number 176 in Veröffentlichungen des Instituts für Wirtschaftsinformatik, pages 15–20. German Research Center for Artificial Intelligence, September 2003.

[Aus62] J. L. Austin. *How to Do Things with Words*. Harvard University Press, Cambridge, Mass., 1962.

[AWM04] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[BA99] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[Bal98] H. Balzert. *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag, 1998.

[BAN03] J. Becker, L. Algermissen, and B. Niehaves. Prozessmodellierung in eGovernment-Projekten mit der eEPK. In M. Nüttgens, F.J. Rump, editor, *Proc. of the 2nd GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2003), Bamberg, Germany*, pages 31–44, 2003.

[BBC+04] P. Bernstein, A. Borgida, M. Carey, S. Chaudhuri, P. Chrysanthis, C. Cliftonand U. Dayal, D. Florescu, M. Franklin, L. Gasser, J. Gehrke, S. Ghandeharizadeh, A. Halevy, H.V. Jagadish, H. Korth, R. Ramakrishnan, A. Rosenthal, S. Sripada, J. Srivastava, J. Su, D. Suciu, S. Thatte, B. Thuraisingham, V. Tsotras, J. Widom, M. Winslett, O. Wolfson, J. Yuan, and M. Zemankova. Science of design for information systems: report of the NSF workshop, Seattle, 2003. *SIGMOD Record*, 33(1):133–137, 2004.

[BD98] E. Badouel and P. Darondeau. Theory of regions. In W. Reisig and G. Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 529–586. Springer, 1998.

[BE05a] U. Brandes and T. Erlebach. Fundamentals. In U. Brandes and T. Erlebach, editors, *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*, volume 3418 of *Lecture Notes in Computer Science*, pages 7–15. Springer, 2005.

[BE05b] U. Brandes and T. Erlebach. Introduction. In U. Brandes and T. Erlebach, editors, *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*, volume 3418 of *Lecture Notes in Computer Science*, pages 1–6. Springer, 2005.

[BE05c] U. Brandes and T. Erlebach, editors. *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*, volume 3418 of *Lecture Notes in Computer Science*. Springer, 2005.

[BEPW03] K. Backhaus, B. Erichson, W. Plinke, and R. Weiber. *Multivariate Analysemethoden. Eine anwendungsorientierte Einführung*. Springer-Verlag, 10th edition, 2003.

[Ber86] G. Berthelot. Checking Properties of Nets Using Transformations. In G. Rozenberg, editor, *Advances in Petri Nets 1985*, volume 222 of *LNCS*, pages 19–40. Springer-Verlag, Berlin, 1986.

[Ber87] G. Berthelot. Transformations and Decompositions of Nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Advances in Petri Nets 1986 Part I: Petri Nets, central models and their properties*, volume 254 of *LNCS*, pages 360–376. Springer-Verlag, Berlin, 1987.

[BG05] S. Balasubramanian and M. Gupta. Structural metrics for goal based business process design and evaluation. *Business Process Management Journal*, 11(6):680–694, 2005.

[BH05] H.U. Buhl and B. Heinrich. Meinung/Dialog: Empirical Research Strategies in Conceptual Modeling - Silver Bullet or Academic Toys? *Wirtschaftsinformatik*, 47(2):152–162, 2005.

[BHK+06] P. Barborka, L. Helm, G. Köldorfer, J. Mendling, G. Neumann, B.F. van Dongen, H.M.W. Verbeek, and W.M.P. van der Aalst. Integration of EPC-related Tools with ProM. In M. Nüttgens and F.J. Rump and J. Mendling, editor, *Proceedings of the 5th GI Workshop on Business Process Management* *with Event-Driven Process Chains (EPK 2006)*, pages 105–120, Vienna, Austria, December 2006. German Informatics Society.

[BK03] J. Becker and M. Kugeler. *Process Management: A Guide for the Design of Business Processes*, chapter The Process in Focus, pages 1–12. Springer-Verlag, 2003.

[BKKR03] M. Bernauer, G. Kappel, G. Kramler, and W. Retschitzegger. Specification of Interorganizational Workflows - A Comparison of Approaches. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*, pages 30–36, 2003.

[BKR03] J. Becker, M. Kugeler, and M. Rosemann. *Process Management: A Guide for the Design of Business Processes*. Springer-Verlag, 2003.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.

[BLN86] C. Batini, M. Lenzerini, and S. B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*, 18(4):323–364, December 1986.

[BN07] J. Becker and B. Niehaves. Epistemological perspectives on is research - a framework for analyzing and systematizing epistemological assumptions. *Information Systems Journal*, 2007. to appear.

[BO02] E. Brabänder and J. Ochs. Analyse und Gestaltung prozessorientierter Risikomanagementsysteme mit Ereignisgesteuerten Prozessketten. In M. Nüttgens and F.J. Rump, editor, *Proc. of the 1st GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2002), Trier, Germany*, pages 17–35, 2002.

[Boe79] B. W. Boehm. *Research Directions in Software Technology*, chapter Software engineering; R & D trends and defense needs. MIT Press, 1979.

[Boe81] B.W. Boehm. *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, 1981.

[BP84] V.R. Basili and B.T. Perricone. Software errors and complexity: An empirical investigation. *Communications of the ACM*, 27(1):42–52, 1984.

[BP98] K. Barkaoui and L. Petrucci. Structural Analysis of Workflow Nets with Shared Resources. In W.M.P. van der Aalst, G. De Michelis, and C.A. Ellis, editors, *Proceedings of Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98)*, volume 98/7 of *Computing Science Reports*, pages 82–95, Lisbon, Portugal, 1998. Eindhoven University of Technology, Eindhoven.

[BR88] V.R. Basili and H. Dieter Rombach. The TAME project: Towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, 14(6):758–773, 1988.

[Bro03] J. vom Brocke. *Referenzmodellierung - Gestaltung und Verteilung von Konstruktionsprozessen*. Number 4 in Advances in Information Systems and Management Science. Logos Verlag Berlin, 2003.

[BRS95] J. Becker, M. Rosemann, and R. Schütte. Grundsätze ordnungsmässiger Modellierung. *Wirtschaftsinformatik*, 37(5):435–445, 1995.

[BRU00] J. Becker, M. Rosemann, and C. von Uthmann. Guidelines of Business Process Modeling. In W.M.P. van der Aalst, J. Desel, and A. Oberweis, editors, *Business Process Management. Models, Techniques, and Empirical Studies*, pages 30–49. Springer, Berlin et al., 2000.

[BS04] J. Becker and R. Schütte. *Handelsinformationssysteme*. Moderne Industrie, Landsberg/Lech, 2nd edition, 2004.

[BS05] M. Brinkmeier and T. Schank. Network statistics. In U. Brandes and T. Erlebach, editors, *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*, volume 3418 of *Lecture Notes in Computer Science*, pages 293–317. Springer, 2005.

[Bun77] M. Bunge. *Treatise on Basic Philosophy. Vol.3. Ontology I. The Furniture of the World.* D. Reidel Publishing, New York, 1977.

[BW84] V.R. Basili and D.M. Weiss. A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, 10(6):728–738, 1984.

[Car05a] J. Cardoso. About the complexity of teamwork and collaboration processes. In *2005 IEEE/IPSJ International Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops), 31 January - 4 February 2005, Trento, Italy*, pages 218–221. IEEE Computer Society, 2005.

[Car05b] J. Cardoso. Control-flow Complexity Measurement of Processes and Weyuker's Properties. In *6th International Enformatika Conference*, Transactions on Enformatika, Systems Sciences and Engineering, Vol. 8, pages 213–218, 2005.

[Car05c] J. Cardoso. Evaluating the process control-flow complexity measure. In *2005 IEEE International Conference on Web Services (ICWS 2005), 11-15 July 2005, Orlando, FL, USA*, pages 803–804. IEEE Computer Society, 2005.

[Car05d] J. Cardoso. *Workflow Handbook 2005*, chapter Evaluating Workflows and Web Process Complexity, pages 284–290. Future Strategies, Inc., Lighthouse Point, FL, USA, 2005.

[Car06] J. Cardoso. Process control-flow complexity metric: An empirical validation. In *Proceedings of IEEE International Conference on Services Computing (IEEE SCC 06), Chicago, USA, September 18-22*, pages 167–173. IEEE Computer Society, 2006.

[CCPP95] F. Casati, S. Ceri, B. Pernici, and G. Pozzi. Conceptual modeling of workflows. In *Proceedings of the OOER International Conference*, Gold Cost, Australia, 1995.

[CFK05] N. Cuntz, J. Freiheit, and E. Kindler. On the semantics of EPCs: Faster calculation for EPCs with small state spaces. In M. Nüttgens and F.J. Rump, editor, *Proceedings of the 4th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2005)*, pages 7–23, Hamburg, Germany, December 2005. German Informatics Society.

[CGK+02] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana. Business Process Execution Language for Web Services, Version 1.0. Specification, BEA Systems, IBM Corp., Microsoft Corp., 2002.

[CGP+05] G. Canfora, F. García, M. Piattini, F. Ruiz, and C.A. Visaggio. A family of experiments to validate metrics for software process models. *Journal of Systems and Software*, 77(2):113–129, 2005.

[Che76] P. Chen. The Entity-Relationship Model - Towards a Unified View of Data. *ACM Transactions on Database Systems (TODS)*, (1):9–36, 1976.

[CK94] S.R. Chidamber and C.F. Kemerer. A metrics suite for object oriented design. *IEEE Transaction on Software Engineering*, 20(6):476–493, 1994.

[CK04] N. Cuntz and E. Kindler. On the semantics of EPCs: Efficient calculation and simulation. In *Proceedings of the 3rd GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2004)*, pages 7–26, 2004.

[CK05] N. Cuntz and E. Kindler. On the semantics of epcs: Efficient calculation and simulation. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume 3649 of *Lecture Notes in Computer Science*, pages 398–403, 2005.

[CKL97] T. Curran, G. Keller, and A. Ladd. *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*. Enterprise Resource Planning Series. Prentice Hall PTR, Upper Saddle River, 1997.

[CKLY98] J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev. Deriving petri nets from finite transition systems. *IEEE Transactions on Computers*, 47(8):859–882, August 1998.

[CLRS01] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[CMNR06] J. Cardoso, J. Mendling, G. Neumann, and H. Reijers. A Discourse on Complexity of Process Models. In Johann Eder and Schahram Dustdar, editors, *Proceedings of BPM Workshops 2006*, volume 4103 of *Lecture Notes in Computer Science*, pages 115–126, Vienna, Austria, 2006. Springer-Verlag.

[Cor98] J. Cortadella. *Petrify: a tutorial for the designer of asychronous circuits*. Universitat Politécnica de Catalunya, http://www.lsi.upc.es/petrify, December 1998.

[CS91] J.C. Cherniavsky and C.H. Smith. On weyuker's axioms for software complexity measures. *IEEE Transactions on Software Engineering*, 17(6):636–638, 1991.

[CS94] R. Chen and A. W. Scheer. Modellierung von Prozessketten mittels Petri-Netz-Theorie. Heft 107, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1994.

[Cun04] N. Cuntz. Über die effiziente Simulation von Ereignisgesteuerten Prozessketten. Master's thesis, University of Paderborn, June 2004. (in German).

[CW98] J.E. Cook and A.L. Wolf. Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.

[DA04] J. Dehnert and W.M.P. van der Aalst. Bridging The Gap Between Business Models And Workflow Specifications. *International J. Cooperative Inf. Syst.*, 13(3):289–332, 2004.

[Dav89] F.D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989.

[Dav93] T. H. Davenport. *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business School Press, Boston, 1993.

[DAV05] B.F. van Dongen, W.M.P. van der Aalst, and H.M.W. Verbeek. Verification of EPCs: Using reduction rules and Petri nets. In O. Pastor and J. Falcão e Cunha, editors, *Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005, Proceedings*, volume 3520 of *Lecture Notes in Computer Science*, pages 372–386. Springer, 2005.

[DE95] J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge Univ. Press, Cambridge, UK, 1995.

[Deh02] J. Dehnert. Making EPCs fit for Workflow Management. In M. Nüttgens and F.J. Rump, editor, *Proc. of the 1st GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2002), Trier, Germany*, pages 51–69, 2002.

[DeM82] T. DeMarco. *Controlling Software Projects*. Yourdon Press, New York, 1982.

[Des05] J. Desel. *Process Aware Information Systems: Bridging People and Software Through Process Technology*, chapter Process Modeling Using Petri Nets, pages 147–178. Wiley Publishing, 2005.

[DFS06] S. Dustdar, J.L. Fiadeiro, and A.P. Sheth, editors. *Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006, Proceedings*, volume 4102 of *Lecture Notes in Computer Science*. Springer, 2006.

[DGR+06] I. Davies, P. Green, M. Rosemann, M. Indulska, and S. Gallo. How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58(3):358–380, 2006.

[DHA05] M. Dumas, A. ter Hofstede, and W.M.P. van der Aalst. *Process Aware Information Systems: Bridging People and Software Through Process Technology*, chapter Introduction, pages 3–20. Wiley Publishing, 2005.

[DHL01] U. Dayal, M. Hsu, and R. Ladin. Business Process Coordination: State of the Art, Trends, and Open Issues. In P.M.G. Apers and P. Atzeni and S. Ceri and S.

Paraboschi and K. Ramamohanarao and R.T. Snodgrass, editor, *Proc. of 27th International Conference on Very Large Data Bases (VLDB), Roma, Italy, Sept. 2001*, 2001.

[DJV05] Boudewijn F. van Dongen and M. H. Jansen-Vullers. Verification of SAP reference models. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume 3649 of *Lecture Notes in Computer Science*, pages 464–469, 2005.

[DMA06] B.F. van Dongen, J. Mendling, and W.M.P. van der Aalst. Structural Patterns for Soundness of Business Process Models. In *Proceedings of EDOC 2006*, Hong Kong, China, 2006. IEEE.

[DMV$^+$05] B.F. van Dongen, A.K. Alves de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The ProM framework: A New Era in Process Mining Tool Support. In G. Ciardo and P. Darondeau, editors, *Application and Theory of Petri Nets 2005*, volume 3536 of *LNCS*, pages 444–454. Springer-Verlag, Berlin, 2005.

[Don07] B.F. van Dongen. *Process Mining and Verification*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2007.

[DPW04] J. Desel, B. Pernici, and M. Weske, editors. *Business Process Management: Second International Conference, BPM 2004, Potsdam, Germany, June 17-18, 2004. Proceedings*, volume 3080 of *Lecture Notes in Computer Science*. Springer, 2004.

[DR01] J. Dehnert and P. Rittgen. Relaxed Soundness of Business Processes. In K.R. Dittrick, A. Geppert, and M.C. Norrie, editors, *Proceedings of the 13th International Conference on Advanced Information Systems Engineering*, volume 2068 of *Lecture Notes in Computer Science*, pages 151–170, Interlaken, 2001. Springer.

[DZ05] J. Dehnert and A. Zimmermann. On the suitability of correctness criteria for business process models. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume 3649 of *Lecture Notes in Computer Science*, pages 386–391, 2005.

[Ell79] C.A. Ellis. Information Control Nets: A Mathematical Model of Office Information Flow. In *Proceedings of the Conference on Simulation, Measurement and Modeling of Computer Systems*, pages 225–240, Boulder, Colorado, 1979. ACM Press.

[EN80] C.A. Ellis and G.J. Nutt. Office information systems and computer science. *ACM Computing Surveys*, 12(1):27–60, 1980.

[EN93] C.A. Ellis and G.J. Nutt. Modelling and Enactment of Workflow Systems. In M. Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993*, volume 691 of *LNCS*, pages 1–16. Springer-Verlag, Berlin, 1993.

[ER89] A. Ehrenfeucht and G. Rozenberg. Partial (Set) 2-Structures - Part 1 and Part 2. *Acta Informatica*, 27(4):315–368, 1989.

[ER05] A. Etien and C. Rolland. Measuring the fitness relationship. *Requir. Eng.*, 10(3):184–197, 2005.

[Esp94] J. Esparza. Reduction and synthesis of live and bounded free choice petri nets. *Information and Computation*, 114(1):50–87, 1994.

[Fay66] H. Fayol. *Administration industrielle et générale. Prévoyance, Organisation, Commandement, Coordination, Control*. Dunod, 1966.

[Fet06] P. Fettke. *Referenzmodellevaluation - Konzeption der strukturalistischen Referenzmodellierung und Entfaltung ontologischer Gütekriterien*. PhD thesis, Universität des Saarlandes, 2006.

[FJJ03] J. Ferraiolo, F. Jun, and D. Jackson. Scalable Vector Graphics (SVG) 1.1. W3C Recommendation 14 January 2003, World Wide Web Consortium, 2003.

[FL03] P. Fettke and P. Loos. Classification of reference models - a methodology and its application. *Information Systems and e-Business Management*, 1(1):35–53, 2003.

[Fla98] R.G. Flatscher. *Meta-Modellierung in EIA/CDIF*. ADV-Verlag, Wien, 1998.

[FO00] N. E. Fenton and N. Ohlsson. Quantitative analysis of faults and failures in a complex software system. *IEEE Transactions on Software Engineering*, 26(8):797–814, August 2000.

[For26] H. Ford. *Today and Tomorrow*. Doubleday, Page and Company, 1926.

[FP97] N.E. Fenton and S.L. Pfleeger. *Software Metrics. A Rigorous and Practical Approach*. PWS, Boston, 1997.

[Fra99] U. Frank. Conceptual Modelling as the Core of Information Systems Discipline - Perspectives and Epistemological Challenges. In *Proceedings of the America Conference on Information Systems - AMCIS '99*, pages 695–698, Milwaukee, 1999.

[Fre79] L.C. Freeman. Centrality in social networks. conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[FS98] O.K. Ferstl and E.J. Sinz. *Grundlagen der Wirtschaftsinformatik Band 1*. R. Oldenbourg Verlag, 3rd edition, 1998.

[FS01] A. Finkel and Ph. Schnoebelen. Well-structured Transition Systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, April 2001.

[FW06] P.J.M. Frederiks and T.P. van der Weide. Information modeling: The process and the required competencies of its participants. *Data & Knowledge Engineering*, 58(1):4–20, 2006.

[GCC+04] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.C. Shan. Business Process Intelligence. *Computers in Industry*, 53(3):321–343, 2004.

[GD05a] A. Selçuk Güceglioglu and O. Demirörs. A process based model for measuring process quality attributes. In Ita Richardson, Pekka Abrahamsson, and Richard Messnarz, editors, *Software Process Improvement, 12th European Conference, EuroSPI 2005, Budapest, Hungary, November 9-11, 2005, Proceedings*, volume 3792 of *Lecture Notes in Computer Science*, pages 118–129. Springer, 2005.

[GD05b] A. Selçuk Güceglioglu and O. Demirörs. Using software quality characteristics to measure business process quality. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume 3649 of *Lecture Notes in Computer Science (LNCS)*, pages 374–379. Springer Verlag, 2005.

[GH03] J.F. Groote and F. van Ham. Large state space visualization. In John Hatcliff Hubert Garavel, editor, *Tools and Algorithms for the Construction and Analysis of Systems, 9th International Conference, TACAS 2003, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003*, volume 2619 of *Lecture Notes in Computer Science*, pages 585–590. Springer-Verlag, 2003.

[GH06] J.F. Groote and F. van Ham. Interactive visualization of large state spaces. *International Journal on Software Tools for Technology Transfer*, 8(1):77–91, 2006.

[GHS95] D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3:119–153, 1995.

[GHW02] G. Guizzardi, H. Herre, and G. Wagner. On the general ontological foundations of conceptual modeling. In S. Spaccapietra, S.T. March, and Y. Kambayashi, editors, *Conceptual Modeling - ER 2002, 21st International Conference on Conceptual Modeling, Tampere, Finland, October 7-11, 2002, Pro-*

ceedings, volume 2503 of *Lecture Notes in Computer Science*, pages 65–78. Springer, 2002.

[Gil88] T. Gilb. *Principles of Software Engineering Management*. Addison-Wesley, Reading, MA, 1988.

[GJM03] C. Ghezzi, M. Jazayeri, and D. Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Pearson Education, Upper Saddle River, 2nd edition, 2003.

[GK91] G.K. Gill and C.F. Kemerer. Cyclomatic complexity density and software maintenance productivity. *IEEE Transaction on Software Engineering*, 17(9):1284–1288, 1991.

[GL07] Volker Gruhn and Ralf Laue. What business process modelers can learn from programmers. *Science of Computer Programming*, 65(1):4–13, 2007.

[GLM06] V. Gruhn, R. Laue, and F. Meyer. Berechnung von Komplexitätsmetriken für ereignisgesteuerte Prozessketten. In M. Nüttgens and F.J. Rump and J. Mendling, editor, *Proceedings of the 5th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2006)*, pages 189–202, Vienna, Austria, December 2006. German Informatics Society.

[Gon86] G. Gong. Cross-validation, the jackknife, and the bootstrap: Excess error estimation in forward logistic regression. *Journal of the American Statistical Association*, 81(393):108–113, March 1986.

[Gro66] E. Grochla. *Automation und Organisation*. Betriebs-wirtschaftlicher Verlag Dr. Th. Gabler, 1966.

[Gro68] E. Grochla. Die Integration der Datenverarbeitung. Durchführung anhand eines integrierten Unternehmensmodells. *Bürotechnik und Automation*, 9:108–120, 1968.

[Gro74] E. Grochla. *Integrierte Gesamtmodelle der Datenverarbeitung. Entwicklung und Anwendung des Kölner Integrationsmodells (KIM)*. Hanser, München - Wien, 1974.

[Gro75] E. Grochla. *Betriebliche Planung und Informationssysteme*. Rowohlt, 1975.

[GRSS05] G. Grossmann, Y. Ren, M. Schrefl, and M. Stumptner. Behavior based integration of composite business processes. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume 3649 of *Lecture Notes in Computer Science*, pages 186–204. Springer, 2005.

[GS75] E. Grochla and N. Szyperski. *Information Systems and Organizational Structure*. Walter de Gruyter, 1975.

[GS95] J. Galler and A.-W. Scheer. Workflow-Projekte: Vom Geschäftsprozeßmodell zur unternehmensspezifischen Workflow-Anwendung. *Information Management*, 10(1):20–27, 1995.

[Gut77] L. Guttman. What is not what in statistics. *The statistician*, 26:81–107, 1977.

[GW03] A. Gemino and Y. Wand. Evaluating modeling techniques based on models of learning. *Commun. ACM*, 46(10):79–84, 2003.

[GW04] A. Gemino and Y. Wand. A framework for empirical evaluation of conceptual modeling techniques. *Requir. Eng.*, 9(4):248–260, 2004.

[GW05] A. Gemino and Y. Wand. Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. *Data & Knowledge Engineering*, 55(3):301–326, 2005.

[Hal77] M. H. Halstead. *Elements of Software Science.*, volume 7 of *Operating, and Programming Systems Series*. Elsevier, Amsterdam, 1977.

[Han70] H.R. Hansen. *Bestimmungsfaktoren fü den Einsatz elektronischer Datenverarbeitungsanlagen in Unternehmungen*. Betriebspolitische Schriften - Beiträge zur Unternehmenspolitik. Duncker & Humblot, Berlin, 1970.

[Har94] A. Hars. *Referenzdatenmodelle. Grundlagen effizienter Datenmodellierung*. Gabler Verlag, 1994.

[HATB98] J. F. Hair, jr., R. E. Anderson, R. L. Tatham, and W. C. Black. *Multivariate Data Analysis*. Prentice-Hall International, Inc., 5th edition edition, 1998.

[Hav05] Mike Havey. *Essential Business Process Modeling*. O'Reilly, August 2005.

[HC93] M. Hammer and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. Harpercollins, New York, 1993.

[Hei96] H. Heilmann. *Information Engineering*, chapter Die Integration der Aufbauorganisation in Workflow-Management-Systemen, pages 147–165. 1996.

[Her00] J. Herbst. A Machine Learning Approach to Workflow Management. In *Proceedings 11th European Conference on Machine Learning*, volume 1810 of *LNCS*, pages 183–194. Springer-Verlag, Berlin, 2000.

[HFKV06] R. Hauser, M. Fries, J.M. Küster, and J. Vanhatalo. Combining analysis of unstructured workflows with transformation to structured workflows. In *Proceedings of EDOC 2006*. IEEE Publishing, October 2006.

[HHK06] B. Hofreiter, C. Huemer, and J.-H. Kim. Choreography of ebxml business collaborations. *Information Systems and E-Business Management*, 2006.

[HHR07] L.J. Heinrich, A. Heinzl, and F. Roithmayr. *Wirtschaftsinformatik. Einfürung und Grundlegung*. R. Oldenbourg Verlag, 3rd edition, 2007.

[HK81] S. Henry and D. Kafura. Software structure metrics based on information-flow. *IEEE Transactions On Software Engineering*, 7(5):510–518, 1981.

[HK89] Rudy Hirschheim and Heinz K. Klein. Four paradigms of information systems development. *Commun. ACM*, 32(10):1199–1216, 1989.

[HK96] M. Hsu and C. Kleissner. Objectflow: Towards a process management infrastructure. *Distributed and Parallel Databases*, 4(2):169–194, 1996.

[HL00] D.W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, 2nd edition, 2000.

[HMPR04] A.R. Hevner, S.T. March, J. Park, and S. Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, March 2004.

[HN05] H.R. Hansen and G. Neumann. *Wirtschaftsinformatik 1: Grundlagen und Anwendungen*. Lucius & Lucius, 9th edition, 2005.

[Hol94] D. Hollingsworth. The Workflow Reference Model. TC00-1003 Issue 1.1, Workflow Management Coalition, 24 November 1994.

[Hol04] D. Hollingsworth. *The Workflow Handbook 2004*, chapter The Workflow Reference Model: 10 Years On, pages 295–312. Workflow Management Coalition, 2004.

[HOS05] K. van Hee, O. Oanea, and N. Sidorova. Colored Petri Nets to Verify Extended Event-Driven Process Chains. In R. Meersman and Z. Tari, editors, *Proceedings of CoopIS/DOA/ODBASE 2005*, volume 3760 of *Lecture Notes in Computer Science*, pages 183–201. Springer-Verlag, 2005.

[HOS+06] K. van Hee, O. Oanea, A. Serebrenik, N. Sidorova, and M. Voorhoeve. Workflow model compositions perserving relaxed soundness. In S. Dustdar, J.L. Fiadeiro, and A. Sheth, editors, *Business Process Management, 4th International Conference, BPM 2006*, volume 4102 of *Lecture Notes in Computer Science*, pages 225–240. Springer-Verlag, September 2006.

[HOSV06] K. van Hee, O. Oanea, N. Sidorova, and M. Voorhoeve. Verifying generalized soundness for workflow nets. In I. Virbitskaite and A. Voronkov, editors, *Proc. of the 6th International Conference on Perspectives of System Informatics, PSI'2006*, volume 4378 of *Lecture Notes in Computer Science*, pages 231–244, Novosibirsk, June 2006. Springer-Verlag.

[HPW05] S. Hoppenbrouwers, H.A. Proper, and T.P. van der Weide. A fundamental view on the process of conceptual modeling. In L.M.L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, and O. Pastor, editors, *Conceptual Modeling - ER 2005, 24th International Conference on Conceptual Modeling, Klagenfurt, Austria,* *October 24-28, 2005, Proceedings*, volume 3716 of *Lecture Notes in Computer Science*, pages 128–143. Springer, 2005.

[HR07] M. Hepp and D. Roman. An ontology framework for semantic business process management. In A. Oberweis, C. Weinhardt, H. Gimpel, A. Koschmider, V. Pankratius, and B. Schnizler, editors, *eOrganisation: Service-, Prozess-, Market Engineering*, pages 423–440. Universitätsverlag Karlsruhe, 2007. Tagungsband der 8. Internationalen Tagung Wirtschaftsinformatik. Band 1.

[HSSV06] K. van Hee, N. Sidorova, L. Somers, and M. Voorhoeve. Consistency in model integration. *Data & Knowledge Engineering*, 56, 2006.

[HSV04] K. van Hee, N. Sidorova, and M. Voorhoeve. Generalised Soundness of Workflow Nets Is Decidable. In J. Cortadella and W. Reisig, editors, *Application and Theory of Petri Nets 2004*, volume 3099 of *LNCS*, pages 197–215. Springer-Verlag, Berlin, 2004.

[HT74] J.E. Hopcroft and R.E. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974.

[HWS93] W. Hoffmann, R. Wein, and A.-W. Scheer. Konzeption eines Steuerungsmodelles für Informationssysteme - Basis für Real-Time-Erweiterung der EPK (rEPK). Heft 106, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1993.

[HWW02] F. van Ham, H. van de Wetering, and J.J. van Wijk. Interactive visualization of state transition systems. *IEEE Transactions on Visualization and Computer Graphics*, 8(4):319–329, 2002.

[IDS01] IDS Scheer AG. *XML-Export und -Import mit ARIS 5.0, January 2001*, 2001.

[IDS03a] IDS Scheer AG. *ARIS 6 Collaborative Suite Methods Manual*, 2003.

[IDS03b] IDS Scheer AG. *XML-Export und -Import (ARIS 6 Collaborative Suite Version 6.2 Schnittstellenbeschreibung).* ftp://ftp.ids-scheer.de/pub/ARIS/HELPDESK/EXPORT/, Juni 2003.

[IEE83] IEEE. *IEEE Standard 729: Glossary of Software Engineering Terminology*. IEEE Computer Society Press, 1983.

[IEE90] IEEE. *IEEE Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology* . IEEE Computer Society Press, 1990.

[ISO91] International Standards Organisation ISO. Information technology - software product evaluation - quality characteristics and guide lines for their use. Iso/iec is 9126, 1991.

[ISR02] ISR. Editorial Statement and Policy. *Information Systems Research*, 13(4):inside front cover, December 2002.

[JB96] S. Jablonski and C. Bussler. *Workflow Management: Modeling Concepts, Architecture, and Implementation*. International Thomson Computer Press, London, UK, 1996.

[Jes06] J. Jeston. *Business Process Management. Practical Guidelines to Successful Implementations*. Butterworth Heinemann, 2006.

[JHG+88] G.G. Judge, R.C. Hill, W.E. Griffiths, H. Lütkepohl, and T.-C. Lee. *Introduction to the theory and practice of econometrics*. John Wiley & Sons, 2nd edition, 1988.

[JKSK00] S. Junginger, H. Kühn, R. Strobl, and D. Karagiannis. Ein geschäftsprozessmanagement-werkzeug der nächsten generation - adonis: Konzeption und anwendungen. *Wirtschaftsinformatik*, 42(5):392–401, 2000.

[JKW07] K. Jensen, L.M. Kristensen, and L. Wells. Coloured Petri nets and CPN tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 2007.

[Joh95] P. Johannesson. Representation and communication - a speech act based approach to information systems design. *Information Systems*, 20(4):291–303, 1995.

[Jon86] C. Jones. *Programmer Productivity*. McGraw-Hill, 1986.

[JVAR06] M. H. Jansen-Vullers, Wil M. P. van der Aalst, and M. Rosemann. Mining configurable enterprise information systems. *Data & Knowledge Engineering*, 56(3):195–244, 2006.

[Kan02] S.H. Kan. *Metrics and Models in Software Quality Engineering*. Addison Wesley, 2nd edition, 2002.

[KBR+05] N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, Y. Lafon, and C. Barreto. Web Services Choreography Description Language Version 1.0. W3C Candidate Recommendation 9 November 2005, World Wide Web Consortium, April 2005.

[Kha04] R.N. Khan. *Business Process Management: A Practical Guide*. Meghan Kiffer, 2004.

[KHB00] B. Kiepuszewski, A.H.M. ter Hofstede, and C. Bussler. On structured workflow modelling. In B. Wangler and L. Bergman, editors, *Advanced Information Systems Engineering, 12th International Conference CAiSE 2000, Stockholm, Sweden, June 5-9, 2000, Proceedings*, volume 1789 of *Lecture Notes in Computer Science*, pages 431–445. Springer, 2000.

[KHHS93] C. Kruse, A. Hars, R. Heib, and A.-W. Scheer. Ways of utilizing reference models for data engineering in CIM. *International Journal of Flexible Automation and Integrated Manufactoring (FAIM)*, 1(1):47–58, 1993.

[KHP+00] S. Kirn, C. Heine, M. Petsch, F. Puppe, F. Klügl, and R. Herrler. Agentenorientierte Modellierung vernetzter Logistikkreisläufe als Ausgangspunkt agentenbasierter Simulation. In S. Kirn and M. Petsch, editors, *Tagungsband zum 2. Kolloquium des DFG-Schwerpunktprogramms "Intelligente Softwareagenten und Betriebswirtschaftliche Anwendungen"*. TU Ilmenau, Lehrstuhl für Wirtschaftsinformatik 2, 2000.

[Kie03] B. Kiepuszewski. *Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows*. PhD thesis, Queensland University of Technology, Brisbane, Australia, 2003.

[Kin03] E. Kindler. On the semantics of EPCs: A framework for resolving the vicious circle (Extended Abstract). In M. Nüttgens, F.J. Rump, editor, *Proc. of the 2nd GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2003), Bamberg, Germany*, pages 7–18, 2003.

[Kin04] E. Kindler. On the semantics of EPCs: A Framework for resolving the vicious circle. In J. Desel and B. Pernici and M. Weske, editor, *Business Process Management, 2nd International Conference, BPM 2004*, volume 3080 of *Lecture Notes in Computer Science*, pages 82–97, 2004.

[Kin06] E. Kindler. On the semantics of EPCs: Resolving the vicious circle. *Data & Knowledge Engineering*, 56(1):23–40, 2006.

[KK02] D. Karagiannis and H. Kühn. Metamodelling Platforms. Invited Paper. In K. Bauknecht and A. Min Tjoa and G. Quirchmayer, editor, *Proceedings of the 3rd International Conference EC-Web 2002 - Dexa 2002, Aix-en-Provence, France*, volume 2455 of *Lecture Notes in Computer Science*, pages 182–196, 2002.

[KLR96] S. Kelly, K. Lyytinen, and M. Rossi. Metaedit+: A fully configurable multi-user and multi-tool case and came environment. In P. Constantopoulos, J. Mylopoulos, and Y. Vassiliou, editors, *Advances Information System Engineering, 8th International Conference, CAiSE'96, Heraklion, Crete, Greece, May 20-24, 1996, Proceedings*, volume 1080 of *Lecture Notes in Computer Science*, pages 1–21. Springer, 1996.

[KLTPZ05] D. Koschützki, K.A. Lehmann, D. Tenfelde-Podehl, and O. Zlotowski. Advanced centrality concepts. In U. Brandes and T. Erlebach, editors, *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*, volume 3418 of *Lecture Notes in Computer Science*, pages 83–111. Springer, 2005.

[KN92] R.S. Kaplan and D.P. Norton. The balanced scorecard - measures that drive performance. *Harvard Business Review*, 70(1):71–79, 1992.

[KN00] R.S. Kaplan and D.P. Norton. Having trouble with your strategy? then map it. *Harvard Business Review*, 78(5):167–176, 2000.

[KNS92] G. Keller, M. Nüttgens, and A.-W. Scheer. Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". Heft 89, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1992.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145, 1995.

[Kos61] E. Kosiol. Modellanalyse als Grundlage unternehmerischer Entscheidungen. *Zeitschrift für betriebswirtschaftlicher Forschung*, pages 318–334, 1961.

[Kos62] E. Kosiol. *Organisation der Unternehmung*. Betriebswirtschaftlicher Verlag Dr. Th. Gabler, 1962.

[KRM06] L.J. Krajewski, L.P. Ritzman, and M.K. Malhotra. *Operations Management. Processes and Value Chains: Process and Value Chains*. Addison-Wesley, 2006.

[KRS05] E. Kindler, V. Rubin, and W. Schäfer. Incremental Workflow mining based on Document Versioning Information. In M. Li, B. Boehm, and L.J. Osterweil, editors, *Proc. of the Software Process Workshop 2005, Beijing, China*, volume 3840 of *LNCS*, pages 287–301. SPRINGER, May 2005.

[KRS06a] E. Kindler, V. Rubin, and W. Schäfer. Activity mining for discovering software process models. In B. Biel, M. Book, and V. Gruhn, editors, *Proc. of the Software Engineering 2006 Conference, Leipzig, Germany*, volume P-79 of *LNI*, pages 175–180. Gesellschaft für Informatik, March 2006.

[KRS06b] E. Kindler, V. Rubin, and W. Schäfer. Process Mining and Petri Net Synthesis. In J. Eder and S. Dustdar, editors, *Business Process Management Workshops*, volume 4103 of *Lecture Notes in Computer Science (LNCS)*, pages 105–116. Springer Verlag, September 2006.

[Kru96] C. Kruse. *Referenzmodellgestütztes Geschäftsprozeßmanagement: Ein Ansatz zur prozeßorientierten Gestaltung betriebslogistischer Systeme*. Gabler Verlag, 1996.

[KSJ06] J. Krogstie, G. Sindre, and H.D. Jørgensen. Process models representing knowledge for action: a revised quality framework. *European Journal of Information Systems*, 15(1):91–102, 2006.

[KT98] G. Keller and T. Teufel. *SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping*. Addison-Wesley, 1998.

[Kug02] M. Kugeler. *Prozessmanagement*, chapter SCM und CRM - Prozessmodellierung für Extende Enterprises, pages 457–485. Springer-Verlag, 3rd edition, 2002.

[Küh06] T. Kühne. Matters of (meta-) modeling. *Software and Systems Modeling*, 5(4):369–385, 2006.

[KUL06] O. Kopp, T. Unger, and F. Leymann. Nautilus Event-driven Process Chains: Syntax, Semantics, and their mapping to BPEL. In M. Nüttgens and F.J. Rump and J. Mendling, editor, *Proceedings of the 5th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2006)*, pages 85–104, Vienna, Austria, December 2006. German Informatics Society.

[LA94] F. Leymann and W. Altenhuber. Managing business processes as an information resource. *IBM Systems Journal*, 33(2):326–348, 1994.

[LA98] P. Loos and T. Allweyer. Process Orientation and Object-Orientation An Approach for Integrating UML and Event-Driven Process Chains (EPC). Heft 144, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1998.

[LB06] L.M. Laird and M.C. Brennan. *Software Measurement and Estimation: A Practical Approach*. IEEE Computer Society and John Wiley & Sons, Inc., Hoboken, New Jersey, 2006.

[Leh97] F. Lehner. *Lexikon der Wirtschaftsinformatik*, chapter Wirtschaftsinformatik, Forschungsgegenstände und Erkenntisverfahren, pages 438–439. Springer-Verlag, 3rd edition, 1997.

[LG06] R. Laue and V. Gruhn. Complexity metrics for business process models. In Witold Abramowicz and Heinrich C. Mayr, editors, *9th International Conference on Business Information Systems (BIS 2006)*, volume 85 of *Lecture Notes in Informatics*, pages 1–12, 2006.

[LK01] A.M. Latva-Koivisto. Finding a complexity for business process models. Research report, Helsinki University of Technology, February 2001.

[LK05] R. Liu and A. Kumar. An analysis and taxonomy of unstructured workflows. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume 3649 of *Lecture Notes in Computer Science*, pages 268–284, 2005.

[LK06] B. List and B. Korherr. An evaluation of conceptual business process modelling languages. In H. Haddad, editor, *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC), Dijon, France, April 23-27, 2006*, pages 1532–1539. ACM, 2006.

[LL00] M. Leuschel and H. Lehmann. Coverability of reset petri nets and other well-structured transition systems by partial deduction. In J.W. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L.M. Pereira, Y. Sagiv, and P.J. Stuckey, editors, *Computational Logic - CL 2000, First International Conference, London, UK, 24-28 July, 2000, Proceedings*, volume 1861 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 2000.

[LLS06] K.C. Laudon, J.P. Laudon, and D. Schoder. *Wirtschaftsinformatik. Eine Einführung*. Pearson Studium, 2006.

[LM04] M. Laguna and J. Marklund. *Business Process Modeling, Simulation and Design*. Prentice Hall, 2004.

[Löw00] J. Löwer. *tDOM A fast XML/DOM/XPath package for Tcl written in C*. http://www.tdom.org/documents/tDOM3.pdf, June 2000. Document to Talk at the First European Tcl/Tk User Meeting.

[LR00] F. Leymann and D. Roller. *Production Workflow - Concepts and Techniques*. Prentice Hall, 2000.

[LS02] K.R. Lang and M. Schmidt. Workflow-supported organizational memory systems: An industrial application. In *35th Hawaii International International Conference on Systems Science (HICSS-35 2002), CD-ROM / Abstracts Proceedings*, page 208. IEEE Computer Society, 2002.

[LSS94] Odd Ivar Lindland, Guttorm Sindre, and Arne Sølvberg. Understanding quality in conceptual modeling. *IEEE Software*, 11(2):42–49, 1994.

[LSW97a] P. Langner, C. Schneider, and J. Wehler. Ereignisgesteuerte Prozeßketten und Petri-Netze. Report Series of the Department of Computer Science 196, University of Hamburg - Computer Sience Department, March 1997. in German.

[LSW97b] P. Langner, C. Schneider, and J. Wehler. Prozeßmodellierung mit ereignisgesteuerten Prozeßketten (EPKs) und Petri-Netzen. *Wirtschaftsinformatik*, 39(5):479–489, 1997.

[LSW98] P. Langner, C. Schneider, and J. Wehler. Petri Net Based Certification of Event driven Process Chains. In J. Desel and M. Silva, editor, *Application and Theory of Petri Nets*, volume 1420 of *Lecture Notes in Computer Science*, pages 286–305, 1998.

[LY90] G.S. Lee and J.-M. Yoon. An empirical study on the complexity metrics of petri nets. In *JTC-CSCC: Joint Technical Conference on Circuits Systems, Computers and Communications, 1990*, pages 327–332, 1990.

[LY92] G.S. Lee and J.-M. Yoon. An empirical study on the complexity metrics of petri nets. *Microelectronics and Reliability*, 32(3):323–329, 1992.

[LZLC02] H. Lin, Z. Zhao, H. Li, and Z. Chen. A novel graph reduction algorithm to identify structural conflicts. In *35th Hawaii International Conference on System Sciences (HICSS-35 2002), CD-ROM / Abstracts Proceedings, 7-10 January 2002, Big Island, HI, USA. Track 9*. IEEE Computer Society, 2002.

[MA06] J. Mendling and W.M.P. van der Aalst. Towards EPC Semantics based on State and Context. In M. Nüttgens and F.J. Rump and J. Mendling, editor, *Proceedings of the 5th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2006)*, pages 25–48, Vienna, Austria, December 2006. German Informatics Society.

[MADV06] J. Mendling, W.M.P. van der Aalst, B.F. van Dongen, and H.M.W. Verbeek. Referenzmodell: Sand im Getriebe - Webfehler. *iX - Magazin für Professionelle Informationstechnik. (in German)*, pages 131–133, August 2006.

[Mar03] A. Martens. On Compatibility of Web Services. *Petri Net Newsletter*, 65:12–20, 2003.

[MB89] T.J. McCabe and C.W. Butler. Design complexity measurement and testing. *Communications of the ACM*, 32:1415–1425, 1989.

[MBK$^+$98] P. Mertens, F. Bodendorf, W. König, A. Picot, and M. Schumann. *Grundzüge der Wirtschaftsinformatik*. Springer-Verlag, 5th edition, 1998.

[McC76] T.J. McCabe. A complexity measure. *IEEE Transactions on Software Engineering*, 2(4):308–320, 1976.

[MCH03] T.W. Malone, K. Crowston, and G.A. Herman, editors. *Organizing Business Knowledge: The Mit Process Handbook: The MIT Process Handbook*. The MIT Press, 2003.

[MDF05] G. Marczyk, D. DeMatteo, and D. Festinger. *Essentials of Research Design and Methodology*. John Wiley & Sons, Inc., 2005.

[MH04] D.L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. W3c recommendation, World Wide Web Consortium, 2004.

[MH05] J. Mendling and M. Hafner. From Inter-Organizational Workflows to Process Execution: Generating BPEL from WS-CDL. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *Proceedings of OTM 2005 Workshops*, volume 3762 of *Lecture Notes in Computer Science*, 2005.

[MLZ05] J. Mendling, K.B. Lassen, and U. Zdun. Transformation strategies between block-oriented and graph-oriented process modelling languages. Technical Report JM-2005-10-10, WU Vienna, October 2005.

[MLZ06a] J. Mendling, K.B. Lassen, and U. Zdun. Experiences in enhancing existing BPM Tools with BPEL Import and Export. In J.L. Fiadeiro S. Dustdar and A. Sheth, editors, *Proceedings of BPM 2006*, volume 4102 of *Lecture Notes in Computer Science*, pages 348–357, Vienna, Austria, 2006. Springer-Verlag.

[MLZ06b] J. Mendling, K.B. Lassen, and U. Zdun. Transformation strategies between block-oriented and graph-oriented process modelling languages. In F. Lehner, H. Nösekabel, and P. Kleinschmidt, editors, *Multikonferenz Wirtschaftsinformatik 2006, XML4BPM Track, Band 2*, pages 297–312, Passau, Germany, February 2006. GITO-Verlag Berlin.

[MMG02] M.L. Markus, A. Majchrzak, and L. Gasser. A Design Theory for Systems that Support Emergent Knowledge Processes. *MIS Quarterly*, 26(3):554–578, September 2002.

[MMN06a] J. Mendling, M. Moser, and G. Neumann. Transformation of yEPC Business Process Models to YAWL. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing*, volume 2, pages 1262–1267, Dijon, France, 2006. ACM.

[MMN+06b] J. Mendling, M. Moser, G. Neumann, H.M.W. Verbeek, B.F. van Dongen, and W.M.P. van der Aalst. A Quantitative Analysis of Faulty EPCs in the SAP Reference Model. BPM Center Report BPM-06-08, BPMCenter.org, 2006.

[MMN+06c] J. Mendling, M. Moser, G. Neumann, H.M.W. Verbeek, B.F. van Dongen, and W.M.P. van der Aalst. Faulty EPCs in the SAP Reference Model. In

J.L. Fiadeiro S. Dustdar and A. Sheth, editors, *Proceedings of BPM 2006*, volume 4102 of *Lecture Notes in Computer Science*, page 451457, Vienna, Austria, 2006. Springer-Verlag.

[MMP05] J. Mendling, M. zur Muehlen, and A. Price. *Process Aware Information Systems: Bridging People and Software Through Process Technology*, chapter Standards for Workflow Definition and Execution, pages 281–316. Wiley Publishing, September 2005.

[MN02] J. Mendling and M. Nüttgens. Event-Driven-Process-Chain-Markup-Language (EPML): Anforderungen zur Definition eines XML-Schemas für Ereignisgesteuerte Prozessketten (EPK). In M. Nüttgens and F.J. Rump, editor, *Proc. of the 1st GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2002), Trier, Germany*, pages 87–93, 2002.

[MN03a] J. Mendling and M. Nüttgens. EPC Modelling based on Implicit Arc Types. In M. Godlevsky and S. W. Liddle and H. C. Mayr, editor, *Proc. of the 2nd International Conference on Information Systems Technology and its Applications (ISTA), Kharkiv, Ukraine*, volume 30 of *Lecture Notes in Informatics*, pages 131–142, 2003.

[MN03b] J. Mendling and M. Nüttgens. EPC Syntax Validation with XML Schema Languages. In M. Nüttgens and F.J. Rump, editor, *Proc. of the 2nd GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2003), Bamberg, Germany*, pages 19–30, 2003.

[MN03c] J. Mendling and M. Nüttgens. XML-basierte Geschäftsprozessmodellierung. In W. Uhr and W. Esswein and E. Schoop, editor, *Proc. of Wirtschaftsinformatik 2003 / Band II, Dresden, Germany*, pages 161 –180, 2003.

[MN04a] J. Mendling and M. Nüttgens. Exchanging EPC Business Process Models with EPML. In M. Nüttgens and J. Mendling, editors, *XML4BPM 2004, Proceedings of the 1st GI Workshop XML4BPM – XML Interchange Formats for Business Process Management at 7th GI Conference Modellierung 2004, Marburg Ger-*

many, pages 61–80, http://wi.wu-wien.ac.at/˜mendling/XML4BPM/xml4bpm-2004-proceedings-epml.pdf, March 2004.

[MN04b] J. Mendling and M. Nüttgens. Transformation of ARIS Markup Language to EPML. In M. Nüttgens and F.J. Rump, editor, *Proceedings of the 3rd GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2004)*, pages 27–38, 2004.

[MN04c] J. Mendling and M. Nüttgens. XML-based Reference Modelling: Foundations of an EPC Markup Language. In J. Becker, editor, *Referenzmodellierung - Proceedings of the 8th GI-Workshop on Reference Modelling, MKWI Essen, Germany*, pages 51–71, 2004.

[MN05] J. Mendling and M. Nüttgens. EPC Markup Language (EPML) - An XML-Based Interchange Format for Event-Driven Process Chains (EPC). Technical Report JM-2005-03-10, Vienna University of Economics and Business Administration, Austria, 2005.

[MN06] J. Mendling and M. Nüttgens. EPC Markup Language (EPML) - An XML-Based Interchange Format for Event-Driven Process Chains (EPC). *Information Systems and e-Business Management*, 4(3):245 – 263, 2006.

[MNN04] J. Mendling, M. Nüttgens, and G. Neumann. A Comparison of XML Interchange Formats for Business Process Modelling. In F. Feltz, A. Oberweis, and B. Otjacques, editors, *Proceedings of EMISA 2004 - Information Systems in E-Business and E-Government*, volume 56 of *Lecture Notes in Informatics*, 2004.

[MNN05a] J. Mendling, G. Neumann, and M. Nüttgens. Towards Workflow Pattern Support of Event-Driven Process Chains (EPC). In M. Nüttgens and J. Mendling, editors, *XML4BPM 2005, Proceedings of the 2nd GI Workshop XML4BPM – XML for Business Process Management at the 11th GI Conference BTW 2005*, volume 145 of *CEUR Workshop Proceedings*, pages 23–38, Karlsruhe, Germany, March 2005. Sun SITE Central Europe.

[MNN05b] J. Mendling, G. Neumann, and M. Nüttgens. *Workflow Handbook 2005*, chapter A Comparison of XML Interchange Formats for Business Process Modelling, pages 185–198. Future Strategies Inc., Lighthouse Point, FL, USA, 2005.

[MNN05c] J. Mendling, G. Neumann, and M. Nüttgens. Yet Another Event-Driven Process Chain. In *Proceedings of BPM 2005*, volume 3649 of *Lecture Notes in Computer Science*, 2005.

[MNN05d] J. Mendling, G. Neumann, and M. Nüttgens. Yet Another Event-driven Process Chain - Modeling Workflow Patterns with yEPCs. *Enterprise Modelling and Information Systems Architectures - an International Journal*, 1(1):3–13, October 2005.

[Moo01] D.L. Moody. *Dealing With Complexity: A Practical Method For Representing Large Entity Relationship Models*. PhD thesis, Department of Information Systems, University of Melbourne, June 2001.

[Moo03] D.L. Moody. Measuring the quality of data models: an empirical evaluation of the use of quality metrics in practice. In *Proceedings of the 11th European Conference on Information Systems, ECIS 2003, Naples, Italy 16-21 June 2003*, 2003.

[Moo05] D.L. Moody. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3):243–276, 2005.

[Mor99] S. Morasca. Measuring attributes of concurrent software specifications in petri nets. In *METRICS '99: Proceedings of the 6th International Symposium on Software Metrics*, pages 100–110, Washington, DC, USA, 1999. IEEE Computer Society.

[MR] J. Mendling and J. Recker. Extending the discussion of model quality: Why clarity and completeness may not be enough. In *Proceedings of the CAiSE*

Workshops at the 19th Conference on Advanced Information Systems Engineering (CAiSE 2007), YEAR =.

[MR04] M. zur Muehlen and M. Rosemann. Multi-Paradigm Process Management. In *Proc. of the Fifth Workshop on Business Process Modeling, Development, and Support - CAiSE Workshops*, 2004.

[MRRA06] J. Mendling, J. Recker, M. Rosemann, and Wil M.P. van der Aalst. Generating Correct EPCs from Configured CEPCs. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing*, volume 2, pages 1505–1511, Dijon, France, 2006. ACM.

[MS95] S.T. March and G. Smith. Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15(4):251–266, December 1995.

[MS06] J. Mendling and C. Simon. Business Process Design by View Integration. In Johann Eder and Schahram Dustdar, editors, *Proceedings of BPM Workshops 2006*, volume 4103 of *Lecture Notes in Computer Science*, pages 55–64, Vienna, Austria, 2006. Springer-Verlag.

[MSBS02] D.L. Moody, G. Sindre, T. Brasethvik, and Arne Sølvberg. Evaluating the quality of process models: Empirical testing of a quality framework. In Stefano Spaccapietra, Salvatore T. March, and Yahiko Kambayashi, editors, *Conceptual Modeling - ER 2002, 21st International Conference on Conceptual Modeling, Tampere, Finland, October 7-11, 2002, Proceedings*, volume 2503 of *Lecture Notes in Computer Science*, pages 380–396. Springer, 2002.

[MT77] F. Mosteller and J. W. Tukey. *Data Analysis and Regression*. Addison-Wesley, 1977.

[Mue04] M. zur Muehlen. *Workflow-based Process Controlling. Foundation, Design, and Implementation of Workflow-driven Process Information Systems.*, volume 6 of *Advances in Information Systems and Management Science*. Logos, Berlin, 2004.

[Mur89] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[MW94] T.J. McCabe and A.H. Watson. Software complexity. *Journal of Defence Software Engineering*, 7(12):5–9, 1994. Crosstalk.

[MZ05a] J. Mendling and J. Ziemann. Transformation of BPEL Processes to EPCs. In M. Nüttgens and F.J. Rump, editor, *Proceedings of the 4th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2005)*, pages 41–53, Hamburg, Germany, December 2005. German Informatics Society.

[MZ05b] J. Mendling and J. Ziemann. Transformation of BPEL Processes to EPCs. In *4th GI Workshop on Event-Driven Process Chains*, Hamburg, Germany, 2005. German Informatics Society.

[Nag91] N.J.D. Nagelkerke. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, September 1991.

[Neu88] G. Neumann. *Metaprogrammierung und Prolog*. Addison-Wesley, December 1988.

[NFM00] N. Fridman Noy, R.W. Fergerson, and M.A. Musen. The knowledge model of protégé-2000: Combining interoperability and flexibility. In R. Dieng and O. Corby, editors, *Knowledge Acquisition, Modeling and Management, 12th International Conference, EKAW 2000, Juan-les-Pins, France, October 2-6, 2000, Proceedings*, volume 1937 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2000.

[Nis94] M.E. Nissen. Valuing it through virtual process measurement. In *Proc. 15th. International Conference on Information Systems, Vancouver, Canada*, pages 309–323, 1994.

[Nis96] M.E. Nissen. *Knowledge-based organizational process redesign: using process flow measures to transform procurement*. PhD thesis, University of South California, 1996.

[Nis98] M.E. Nissen. Redesigning reengineering through measurement-driven inference. *MIS Quarterly*, 22(4):509–534, 1998.

[Nor32] F. Nordsieck. *Die Schaubildliche Erfassung und Untersuchung der Betriebsorganisation*. Organisation - Eine Schriftenreihe. C. E. Poeschel Verlag, Stuttgart, 1932.

[Nor34] F. Nordsieck. *Grundlagen der Organisationslehre*. C. E. Poeschel Verlag, 1934.

[NPW03] S. Neumann, C. Probst, and C. Wernsmann. Continuous Process Management. In J. Becker, M. Kugeler, and M. Rosemann, editors, *Process Management: A Guide for the Design of Business Processes*, pages 233–250. Springer, Berlin, New York, 2003.

[NR02] M. Nüttgens and F.J. Rump. Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In J. Desel and M. Weske, editor, *Proceedings of Promise 2002, Potsdam, Germany*, volume 21 of *Lecture Notes in Informatics*, pages 64–77, 2002.

[Nüt95] M. Nüttgens. *Koordiniert-dezentrales Informationsmanagement: Rahmenkonzept, Koordinationsmodelle und Werkzeug-Shell*. PhD thesis, Rechts- und Wirtschaftswissenschaftliche Fakultät der Universität des Saarlandes, 1995.

[NZ00] G. Neumann and U. Zdun. XOTcl, an Object-Oriented Scripting Language. In *Proc. of the 7th USENIX Tcl/Tk Conference, Austin, Texas, USA*, 2000.

[Obe96] A. Oberweis. An integrated approach for the specification of processes and related complex structured objects in business applications. *Decision Support Systems*, 17:31–53, 1996.

[ÖG92] H. Österle and T. Gutzwiller. *Konzepte angewandter Analyse- und Design-Methoden. Band 1: Ein Referenz-Metamodell für die Analyse und das System-Design*. Angewandte InformationsTechnik-Verl. GmbH, 1992.

[OMG02] OMG, ed. Meta Object Facility. Version 1.4, Object Management Group, 2002.

[OMG04] OMG, ed. Unified Modeling Language. Version 2.0, Object Management Group, 2004.

[OMG06] OMG, ed. Business Process Modeling Notation (BPMN) Specification. Final Adopted Specification, dtc/06-02-01, Object Management Group, February 2006.

[OS96] A. Oberweis and P. Sander. Information system behavior specification by high-level petri nets. *ACM Transactions on Information Systems*, 14(4):380–420, 1996.

[Öst95] H. Österle. *Business Engineering Prozeß- und Systementwicklung. Band 1: Entwurfstechniken*. Springer Verlag, 1995.

[Pal07] Nathaniel Palmer. A survey of business process initiatives. BPT Report, Business Process Trends and Transformation+Innovation, January 2007.

[Pem02] S. Pemberton et al. XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). W3C Recommendation 26 January 2000, revised 1 August 2002, World Wide Web Consortium, 2002.

[Pet62] C.A. Petri. *Kommunikation mit Automaten*. PhD thesis, Fakultät für Mathematik und Physik, Technische Hochschule Darmstadt, Darmstadt, Germany, 1962.

[PH07] S. Philippi and H.J. Hill. Communication support for systems engineering - process modelling and animation with april. *The Journal of Systems and Software,*, 2007. accepted for publication.

[Pnu77] A. Pnueli. The Temporal Logic of Programs. In *Proceedings of the 18th IEEE Annual Symposium on the Foundations of Computer Science*, pages 46–57. IEEE Computer Society Press, Providence, 1977.

[Por85] M. E. Porter. *Competitive Advantage : Creating and Sustaining Superior Performance*. The Free Press, New York, 1985.

[Pri95] J. Priemer. *Entscheidungen über die Einsetzbarkeit von Software anhand formaler Modelle*. PhD thesis, Westfälische Wilhelms-Universität Münster, 1995.

[PW05] A.J. Pretorius and J.J. van Wijk. Multidimensional visualization of transition systems. In *9th International Conference on Information Visualisation, IV 2005, 6-8 July 2005, London, UK*, pages 323–328. IEEE Computer Society, 2005.

[PW06a] A.J. Pretorius and J.J. van Wijk. Visual analysis of multivariate state transition graphs. *IEEE Visualization and Computer Graphics*, 12(5):685–692, 2006.

[PW06b] F. Puhlmann and M. Weske. Investigations on soundness regarding lazy activities. In S. Dustdar, J.L. Fiadeiro, and A. Sheth, editors, *Business Process Management, 4th International Conference, BPM 2006*, volume 4102 of *Lecture Notes in Computer Science*, pages 145–160. Springer-Verlag, September 2006.

[RA07] M. Rosemann and Wil van der Aalst. A Configurable Reference Modelling Language. *Information Systems*, 32:1–23, 2007.

[RD98] M. Reichert and P. Dadam. ADEPTflex: Supporting Dynamic Changes of Workflow without Loosing Control. *Journal of Intelligent Information Systems*, 10(2):93–129, 1998.

[Rei03] H.A. Reijers. A cohesion metric for the definition of activities in a workflow process. In *Proceedings of the Eighth CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2003)*, pages 116–125, 2003.

[RG02] M. Rosemann and P. Green. Developing a meta model for the Bunge-Wand-Weber ontological constructs. *Information Systems*, 27:75–91, 2002.

[RGA+06] V. Rubin, C.W. Günther, W.M.P. van der Aalst, E. Kindler, B.F. van Dongen, and W. Schäfer. Process mining framework for software processes. BPMCenter Report BPM-07-01, BPMcenter.org, 2006.

[Rit99] P. Rittgen. Modified EPCs and Their Formal Semantics. Arbeitsberichte des Instituts für Wirtschaftsinformatik 19, Universität Koblenz-Landau, 1999.

[Rit00] P. Rittgen. Paving the Road to Business Process Automation. In *Proc. of the European Conference on Information Systems (ECIS), Vienna, Austria*, pages 313–319, 2000.

[RMRA06] J. Recker, J. Mendling, M. Rosemann, and W.M.P. van der Aalst. Model-driven Enterprise Systems Configuration. In *Proceedings of the 18th Conference on Advanced Information Systems Engineering (CAiSE 2006)*, volume 4001 of *Lecture Notes in Computer Science*, pages 369–383, Luxembourg, Luxembourg, 2006. Springer-Verlag.

[Rod02] J. Rodenhagen. Ereignisgesteuerte Prozessketten - Multi-Instantiierungsfähigkeit und referentielle Persistenz. In *Proceedings of the 1st GI Workshop on Business Process Management with Event-Driven Process Chains*, pages 95–107, 2002.

[Ros95] M. Rosemann. *Erstellung und Integration von Prozeßmodellen - Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung*. PhD thesis, Westfälische Wilhelms-Universität Münster, 1995.

[Ros03] M. Rosemann. *Process Management: A Guide for the Design of Business Processes*, chapter Preparation of Process Modeling, pages 41–78. Springer-Verlag, 2003.

[Ros06] M. Rosemann. Potential pitfalls of process modeling: part a. *Business Process Management Journal*, 12(2):249–254, 2006.

[Rum99] F.J. Rump. *Geschäftsprozessmanagement auf der Basis ereignisgesteuerter Prozessketten - Formalisierung, Analyse und Ausführung von EPKs*. Teubner Verlag, 1999.

[RV04] H.A. Reijers and I.T.P. Vanderfeesten. Cohesion and coupling metrics for workflow process design. In J. Desel, B. Pernici, and M. Weske, editors, *Business Process Management: Second International Conference, BPM 2004, Potsdam, Germany, June 17-18, 2004. Proceedings*, volume 3080 of *Lecture Notes in Computer Science*, pages 290–305. Springer, 2004.

[SA05] S.M. Smith and G.S. Albaum. *Fundamentals of Marketing Research*. Sage Publications, 2005.

[SAJ+02] E. Söderström, B. Andersson, P. Johannesson, E. Perjons, and B. Wangler. Towards a Framework for Comparing Process Modelling Languages. In A. Banks Pidduck, J. Mylopoulos, C.C. Woo, and M.T. Özsu, editors, *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE)*, volume 2348 of *Lecture Notes in Computer Science*, pages 600–611, 2002.

[SB91] R.W. Selby and V.R. Basili. Analyzing error-prone system structure. *IEEE Transactions on Software Engineering*, 17(2):141–152, 1991.

[Sch76] A.-W. Scheer. *Produktionsplannung auf der Grundlage einer Datenbank des Fertigungsbereichs*. Oldenbourg, 1976.

[Sch78] A.-W. Scheer. *Wirtschafts- und Betriebsinformatik*. Verlag Moderne Industrie, München, 1978.

[Sch84] A.-W. Scheer. *EDV-orientierte Betriebswirtschaftslehre*. Springer-Verlag, Berlin, Heidelberg, New York, Tokio, 1984.

[Sch87] A.-W. Scheer. *CIM - Computer integrated manufacturing : der computergesteuerte Industriebetrieb*. Springer-Verlag, 1987.

[Sch98a] A.-W. Scheer. *ARIS - Business Process Frameworks*. Springer, Berlin et al., 2nd edition, 1998.

[Sch98b] A.-W. Scheer. *Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse*. Springer-Verlag, 2nd edition edition, 1998.

[Sch98c] R. Schütte. *Grundsätze ordnungsgemäßer Referenzmodellierung*. Gabler Verlag, 1998.

[Sch00] A.-W. Scheer. *ARIS - Business Process Modeling*. Springer, Berlin et al., 3rd edition, 2000.

[Sch02] A.-W. Scheer. *ARIS in der Praxis - Gestaltung, Implementierung und Optimierung von Geschäftsprozessen*, chapter ARIS: Von der Vision zur praktischen Geschäftsprozesssteuerung, pages 1–14. Springer-Verlag, 2002.

[SCJ06] N. Slack, S. Chambers, and R. Johnston. *Operations and Process Management. Principles and Practice for Strategic Impact*. Prentice Hall, 2006.

[Sco00] J. Scott. *Social Network Analysis: A Handbook*. SAGE Publications Ltd, 2nd edition, 2000.

[SD95] R. Striemer and W. Deiters. Workflow Management - Erfolgreiche Planung und Durchführung von Workflow-Projekten. Arbeitsbericht, Fraunhofer Institut für Software- und Systemtechnik, 1995.

[SDL05] K. Sarshar, P. Dominitzki, and P. Loos. Einsatz von Ereignisgesteuerten Prozessketten zur Modellierung von Prozessen in der Krankenhausdomäne Eine empirische Methodenevaluation. In M. Nüttgens and F.J. Rump, editor, *Proceedings of the 4th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2005)*, pages 97–116, Hamburg, Germany, December 2005. German Informatics Society.

[Sea69] J. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, Eng., 1969.

[SEI92] Software Engineering Institute SEI. Software size measurement: A framework for counting source statements. Technical Report CMU/SEI-92-TR-020, ADA258304, 1992.

[Sei02] H. Seidlmeier. *Prozessmodellierung mit ARIS*. Vieweg Verlag, 2002.

[SF06] H. Smith and P. Fingar. *Business Process Management: The Third Wave*. Meghan Kiffer, 2006.

[SH05] P. Stahlknecht and U. Hasenkamp. *Einführung in die Wirtschaftsinformatik*. Springer-Verlag, 11th edition, 2005.

[Sha88] S.M. Shatz. Towards complexity metrics for ada tasking. *IEEE Transaction on Software Engineering*, 14(8):1122–1127, 1988.

[Sil01a] L. Silverston. *The Data Model Resource Book, Volume 1, A Library of Universal Data Models for all Enterprises*. John Wiley and Sons, New York, revised edition, 2001.

[Sil01b] L. Silverston. *The Data Model Resource Book, Volume 2, A Library of Data Models for Specific Industries*. John Wiley and Sons, New York, revised edition, 2001.

[Sim96] H.A. Simon. *Sciences of the Artificial*. The MIT Press, 3rd edition, 1996.

[Sim06] Carlo Simon. *Negotiation Processes. The semantic process language and applications*. University of Koblenz, September 2006. Habilitationsschrift.

[SL05] K. Sarshar and P. Loos. Comparing the control-flow of epc and petri net from the end-user perspective. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, LNCS 3649, pages 434–439, 2005.

[SLTM91] K. Smolander, K. Lyytinen, V.-P. Tahvanainen, and P. Marttiin. Metaedit - a flexible graphical environment for methodology modelling. In R. Andersen, J.A. Bubenko Jr., and A. Sølvberg, editors, *Advanced Information Systems Engineering, CAiSE'91, Trondheim, Norway, May 13-15, 1991, Proceedings*, volume 498 of *Lecture Notes in Computer Science*, pages 168–193. Springer, 1991.

[SM06] C. Simon and J. Mendling. Verification of Forbidden Behavior in EPCs. In H.C. Mayr and R. Breu, editors, *Proceedings of the GI Conference Modellierung*, volume 82 of *Lecture Notes in Informatics*, pages 233–242, Innsbruck, Austria, March 2006. German Informatics Society.

[Smi76] A. Smith. *An inquiry into the nature and causes of the wealth of nations*. London, 1776.

[Smi07] R.F. Smith. *Business Process Management and the Balanced Scorecard. Focusing Processes as Strategic Drivers: Using Processes as Strategic Drivers*. Wiley & Sons, 2007.

[SNZ97] A.-W. Scheer, M. Nüttgens, and V. Zimmermann. Objektorientierte Ereignisgesteuerte Prozeßketten (oEPK) - Methode und Anwendung. Heft 141, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1997.

[SO96] W. Sadiq and M.E. Orlowska. Modeling and verification of workflow graphs. Technical Report No. 386, Department of Computer Science, The University of Queensland, Australia, 1996.

[SO99] W. Sadiq and M.E. Orlowska. Applying graph reduction techniques for identifying structural conflicts in process models. In M. Jarke and A. Oberweis, editors, *Advanced Information Systems Engineering, 11th International Conference CAiSE'99, Heidelberg, Germany, June 14-18, 1999, Proceedings*, volume 1626 of *Lecture Notes in Computer Science*, pages 195–209. Springer, 1999.

[SO00] W. Sadiq and M.E. Orlowska. Analyzing Process Models using Graph Reduction Techniques. *Information Systems*, 25(2):117–134, 2000.

[Som01] I. Sommerville. *Software Engineering*. Addison-Wesley, 6th edition, 2001.

[SR98] R. Schütte and T. Rotthowe. The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models. In Tok Wang Ling, Sudha Ram, and Mong-Li Lee, editors, *Proceedings of the 17th International Conference* *on Conceptual Modeling*, volume 1507 of *Lecture Notes in Computer Science*, pages 240–254, Singapore, 1998. Springer.

[SRS96] B. Scholz-Reiter and E. Stickel, editors. *Business Process Modelling*. Springer-Verlag, 1996.

[ST03] K. Schneider and O. Thomas. Kundenorientierte Dienstleistungsmodellierung mit Ereignisgesteuerten Prozessketten. In M. Nüttgens and F.J. Rump, editor, *Proc. of the 2nd GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2003), Bamberg, Germany*, pages 87–93, 2003.

[Sta73] H. Stachowiak. *Allgemeine Modelltheorie*. Springer-Verlag, 1973.

[STA05] A.-W. Scheer, O. Thomas, and O. Adam. *Process Aware Information Systems: Bridging People and Software Through Process Technology*, chapter Process Modeling Using Event-Driven Process Chains, pages 119–146. Wiley Publishing, 2005.

[Sta06] J.L. Staud. *Geschäftsprozessanalyse: Ereignisgesteuerte Prozessketten und Objektorientierte Geschäftsprozessmodellierung für Betriebswirtschaftliche Standardsoftware*. Springer-Verlag, 3rd edition, July 2006.

[Ste46] S.S. Stevens. On the theory of scale types and measurement. *Science*, 103:677–680, 1946.

[Ste51] S.S. Stevens. *Handbook of Experimental Psychology*, chapter Mathematics, Measurement, and Psychophysics. John Wiley, 1951.

[Sto74] M. Stone. Cross validity choice and assessment of statistical predictors. *Journal of the Royal Statistical Society*, 36:111–147, 1974.

[Str96] S. Strahringer. *Metamodellierung als Instrument des Methodenvergleichs. Eine Evaluierung am Beispiel objektorientierter Analysemethoden*. Shaker Verlag, Aachen, 1996.

[SW63] C.E. Shannon and W. Weaver. *Mathematical Theory of Communication*. B&T, 1963.

[SW04] P. Soffer and Y. Wand. Goal-driven analysis of process model validity. In A. Persson and J. Stirna, editors, *Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004, Riga, Latvia, June 7-11, 2004, Proceedings*, volume 3084 of *Lecture Notes in Computer Science*, pages 521–535. Springer, 2004.

[Tay11] F.W. Taylor. *The Principles of Scientific Management*. Harper and Brothers, New York and London, 1911.

[TD06a] O. Thomas and T. Dollmann. Fuzzy-EPK-Modelle: Attributierung und Regelintegration. In M. Nüttgens and F.J. Rump and J. Mendling, editor, *Proceedings of the 5th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2006)*, pages 49–68, Vienna, Austria, December 2006. German Informatics Society.

[TD06b] W. Trochim and J.P. Donnelly. *The Research Methods Knowledge Base*. Atomic Dog Publishing, 3rd edition, 2006.

[TF06a] O. Thomas and M. Fellmann. Semantic event-driven process chains. In *Semantics for Business Process Management 2006 (SBPM 2006) : Workshop at 3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, June 2006.

[TF06b] O. Thomas and M. Fellmann. Semantische ereignisgesteuerte prozessketten. In J. Schelp, R. Winter, U. Frank, B. Rieger, and K. Turowski, editors, *Integration, Informationslogistik und Architektur: DW2006*, 2006.

[THA02] O. Thomas, C. Hüsselmann, and O. Adam. Fuzzy-Ereignisgesteuerte Prozessketten - Geschäftsprozessmodellierung unter Berücksichtigung unscharfer Daten. In M. Nüttgens and F.J. Rump, editor, *Proc. of the 1st GI-Workshop on Business Process Management with Event-Driven Process Chains (EPK 2002), Trier, Germany*, pages 7–16, 2002.

[Tho05] O. Thomas. Understanding the term reference model in information systems research: History, literature analysis and explanation. In C. Bussler and

A. Haller, editors, *Business Process Management Workshops, BPM 2005 International Workshops, BPI, BPD, ENEI, BPRM, WSCOBPM, BPS, Nancy, France, September 5, 2005, Revised Selected Papers*, volume 3812 of *Lecture Notes in Computer Science*, pages 484–496, 2005.

[Tja01] G.S. Tjaden. Business process structural analysis. Technical report, Georgia Tech Research Corp., June 2001.

[TM05] F.L. Tiplea and D.C. Marinescu. Structural soundness of workflow nets is decidable. *Inf. Process. Lett.*, 96(2):54–58, 2005.

[TNM96] G.S. Tjaden, S. Narasimhan, and S. Mitra. Structural effectiveness metrics for business processes. In *Proceedings of the INFORMS Conference on Information Systems and Technology*, pages 396–400, May 1996.

[Too04] R. van der Toorn. *Component-Based Software Design with Petri nets: An Approach Based on Inheritance of Behavior*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2004.

[Tor58] W.S. Torgerson. *Theory and methods of scaling*. Wiley, 1958.

[TS06] O. Thomas and A.-W. Scheer. Tool support for the collaborative design of reference models - a business engineering perspective. In *39th Hawaii International International Conference on Systems Science (HICSS-39 2006), CD-ROM / Abstracts Proceedings, 4-7 January 2006, Kauai, HI, USA*. IEEE Computer Society, 2006.

[Tuk77] J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[Ulr49] H. Ulrich. *Betriebswirtschaftliche Organisationslehre. Eine Einführung*. Verlag Paul Haupt, 1949.

[VA00] H.M.W. Verbeek and W.M.P. van der Aalst. Woflan 2.0: A Petri-net-based Workflow Diagnosis Tool. In M. Nielsen and D. Simpson, editors, *Application and Theory of Petri Nets 2000*, volume 1825 of *LNCS*, pages 475–484. Springer-Verlag, Berlin, 2000.

[VA06] H. M. W. Verbeek and W.M.P. van der Aalst. On the verification of EPCs using T-invariants. BPMCenter Report BPM-06-05, BPMcenter.org, 2006.

[Val98] A. Valmari. The state explosion problem. In W. Reisig and G. Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 429–528. Springer, 1998.

[VBA01] H.M.W. Verbeek, T. Basten, and W.M.P. van der Aalst. Diagnosing Workflow Processes using Woflan. *The Computer Journal*, 44(4):246–279, 2001.

[VDMA06] H.M.W. Verbeek, B.F. van Dongen, J. Mendling, and W.M.P. van der Aalst. Interoperability in the ProM Framework. In T. Latour and M. Petit, editors, *Proceedings of the CAiSE Workshops at the 18st Conference on Advanced Information Systems Engineering (CAiSE 2006)*, pages 619–630, Luxembourg, Luxembourg, 2006.

[Ver04] H.M.W. Verbeek. *Verification and Enactment of Workflow Management Systems*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2004.

[VPAJ07] H.M.W. Verbeek, A.J. Pretorius, W.M.P. van der Aalst, and J.J. van Wijk. Visualizing state spaces with Petri nets. Computer Science Report 07/01, Eindhoven University of Technology, Eindhoven, The Netherlands, 2007.

[VPL07] V. K. Vaishnavi, S. Purao, and J. Liegle. Object-oriented product metrics: A generic framework. *Information Sciences*, 177:587–606, January 2007.

[VW93] P.F. Velleman and L. Wilkinson. Nominal, ordinal, interval, and ratio typologies are misleading. *The American Statistician*, 47(1):65–72, February 1993.

[WAD$^+$05] P. Wohed, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, and N. Russell. Pattern-Based Analysis of the Control-Flow Perspective of UML Activity Diagrams. In L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, and

O. Pastor, editors, *24nd International Conference on Conceptual Modeling (ER 2005)*, volume 3716 of *LNCS*, pages 63–78. Springer-Verlag, Berlin, 2005.

[WAD+06] P. Wohed, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, and N. Russell. On the Suitability of BPMN for Business Process Modelling. In S. Dustdar, J.L. Fiadeiro, and A. Sheth, editors, *Business Process Management, 4th International Conference, BPM 2006*, volume 4102 of *Lecture Notes in Computer Science*, pages 161–176. Springer-Verlag, September 2006.

[WADH03] P. Wohed, Wil M.P. van der Aalst, M. Dumas, and Arthur H. M. ter Hofstede. Analysis of Web Service Composition Languages: The Case of BPEL4WS. In *Proceedings of Conceptual Modeling - ER 2003*, volume LNCS 2813, pages 200–215, 2003.

[WAHE06] M.T. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Verifying Workflows with Cancellation Regions and OR-joins: An Approach Based on Reset Nets and Reachability Analysis. In S. Dustdar, J.L. Fiadeiro, , and A. Sheth, editors, *Proceedings of BPM 2006*, volume 4102 of *Lecture Notes in Computer Science*, pages 389–394, Vienna, Austria, 2006. Springer-Verlag.

[WEAH05] M.T. Wynn, D. Edmond, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Achieving a General, Formal and Decidable Approach to the OR-join in Workflow using Reset nets. In G. Ciardo and P. Darondeau, editors, *Applications and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 423–443, 2005.

[Wes00] M. Weske. Foundation, Design, and Implementation of Dynamic Adaptations in a Workflow Management System. Fachbericht Angewandte Mathematik und Informatik 6/2000-I, Universität Münster, Münster, Germany, 2000.

[Wey88] E.J. Weyuker. Evaluating software complexity measures. *IEEE Transactions on Software Engineering*, 14(9):1357–1365, 1988.

[WHM06] I. Weber, J. Haller, and J.A. Mülle. Derivation of executable business processes from choreographies in virtual organizations. In *Proc. of XML4BPM 2006*, February 2006.

[Win06] Wintergreen Research. Business process management (bpm) market opportunities, strategies, and forecasts, 2006 to 2012. Technical Report Pub ID: WGR1352720, Wintergreen Research, http://www.marketresearch.com/product/display.asp?productid=1352720&g=1, October 2006.

[Win88] T. Winograd. A language/action perspective on the design of cooperative work. *Human-Computer Interaction*, 3(1):3–30, 1987-88.

[WJH03] B.B. Welch, K. Jones, and J. Hobbs. *Practical Programming in Tcl and Tk*. Prentice Hall International, 4th edition, 2003.

[WKR01] A. Winter, B. Kullbach, and V. Riediger. An Overview of the GXL Graph Exchange Language. In S. Diehl, editor, *Software Visualization - International Seminar Dagstuhl Castle*, volume 2269 of *Lecture Notes in Computer Science*, pages 324–336, 2001.

[WLPS06] U. Wahli, L. Leybovich, E. Prevost, and R. Scher. *Business Process Management: Modeling Through Monitoring*. IBM Redbooks. Vervante, 2006.

[Wor02] Workflow Management Coalition. Workflow Process Definition Interface – XML Process Definition Language. Document Number WFMC-TC-1025, October 25, 2002, Version 1.0, Workflow Management Coalition, 2002.

[Wor05] Workflow Management Coalition. Workflow Process Definition Interface – XML Process Definition Language. Document Number WFMC-TC-1025, October 3, 2005, Version 2.00, Workflow Management Coalition, 2005.

[WVA$^+$06a] M.T. Wynn, H.M.W. Verbeek, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Reduction rules for reset workflow nets. BPMCenter Report BPM-06-25, BPMcenter.org, 2006.

[WVA+06b] M.T. Wynn, H.M.W. Verbeek, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Reduction rules for yawl workflow nets with cancellation regions and or-joins. BPMCenter Report BPM-06-24, BPMcenter.org, 2006.

[WW90] Y. Wand and R. Weber. *Studies in Bunge's Treatise on Basic Philosophy*, chapter Mario Bunge's Ontology as a Formal Foundation for Information Systems Concepts, pages 123–149. the Poznan Studies in the Philosophy of the Sciences and the Humanities. Rodopi, 1990.

[WW95] Y. Wand and R. Weber. On the deep structure of information systems. *Information Systems Journal*, 5:203–223, 1995.

[WW02] Y. Wand and R. Weber. Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research*, 13(4):363–376, 2002.

[YC79] E. Yourdon and L.L. Constantine. *Structured Design*. Prentice Hall, 1979.

[ZHB+06] W. Zhao, R. Hauser, K. Bhattacharya, B.R. Bryant, and F. Cao. Compiling business processes: untangling unstructured loops in irreducible flow graphs. *Int. Journal of Web and Grid Services*, 2(1):68–91, 2006.

[Zis77] M.D. Zisman. *Representation, Specification and Automation of Office Procedures*. PhD thesis, University of Pennsylvania, Warton School of Business, 1977.

[Zis78] M. D. Zisman. Use of production systems for modeling asynchronous concurrent processes. *Pattern-Directed Inference Systems*, pages 53–68, 1978.

[ZM05] J. Ziemann and J. Mendling. EPC-Based Modelling of BPEL Processes: a Pragmatic Transformation Approach. In *International Conference "Modern Information Technology in the Innovation Processes of the Industrial Enterprises*, Genova, Italy, 2005.

[ZR96] O. Zukunft and F.J. Rump. *Business Process Modelling*, chapter From Business Process Modelling to Workflow Management: An Integrated Approach, pages 3–22. Springer-Verlag, 1996.

[Zus91] H. Zuse. *Software Complexity: Measures and Methods*. Walter de Gruyter and Co, New Jersey, 1991.