SUPPORTING SOFTWARE PROCESSES ANALYSIS AND DECISION-MAKING USING PROVENANCE DATA

Gabriella Castro Barbosa Costa Dalpra

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadoras: Cláudia Maria Lima Werner
Regina Maria Maciel Braga Villela

Rio de Janeiro
Outubro de 2018

SUPPORTING SOFTWARE PROCESSES ANALYSIS AND DECISION-MAKING USING PROVENANCE DATA

Gabriella Castro Barbosa Costa Dalpra

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

_____
Prof.ª Cláudia Maria Lima Werner, D.Sc.

_____
Prof.ª Regina Maria Maciel Braga Villela, D.Sc.

_____
Prof. Toacy Cavalcante de Oliveira, D.Sc.

_____
Prof.ª Marta Lima de Queiros Mattoso, D.Sc.

_____
Prof. Leonardo Gresta Paulino Murta, D.Sc.

_____
Prof. Ricardo de Almeida Falbo, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
OUTUBRO DE 2018

Dalpra, Gabriella Castro Barbosa Costa

Supporting Software Processes Analysis and Decision-Making Using Provenance Data/ Gabriella Castro Barbosa Costa Dalpra. – Rio de Janeiro: UFRJ/COPPE, 2018.

XIII, 201 p.: il.; 29,7 cm.

Orientadora: Cláudia Maria Lima Werner

Regina Maria Maciel Braga Villela

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2018.

Referências Bibliográficas: p. 123-131.

1. Software Processes. 2. Software Development Processes. 3. Provenance Data. 4. Software Engineering. I. Werner, Cláudia Maria Lima *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

“So the problem is not so much to see what nobody has yet seen, as to think what nobody has yet thought concerning that which everybody sees.”

(Arthur Schopenhauer)

ACKNOWLEDGMENTS

I am deeply grateful to God, for the many experiences that I have had during these five years of my PhD course. It was not easy (definitely!), but it was amazing!

Claudia and Regina, you are my inspiration to move forward in the academic career. I will be eternally grateful for all the teachings, advices, and opportunities! What I learned from you goes much beyond the work presented in this thesis and I am not be able to express in words how much you are important to me.

I am so grateful for all the members of my lovely family (in special for my father José Wilson, my mother Heloiza, and my brother Fellipe) for supporting me during this trajectory.

Humberto, thank you for being by my side until the end! I do not know if I could be here without you. Only you know what we've been through here... I dedicate all this work to you!

Last, but not least, I must say that many whom I am grateful are not mentioned here. I confess that I have no more 'strength' to write, after these more than two hundred pages. You know how grateful I am, and you will be always in my heart!

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

APOIANDO A ANÁLISE E A TOMADA DE DECISÃO EM PROCESSOS DE SOFTWARE USANDO DADOS DE PROVENIÊNCIA

Gabriella Castro Barbosa Costa Dalpra

Outubro/2018

Orientadoras: Cláudia Maria Lima Werner
Regina Maria Maciel Braga Villela

Programa: Engenharia de Sistemas e Computação

Proveniência de dados é definida como a descrição da origem de um dado e o processo pelo qual este passou até chegar ao seu estado atual. Proveniência de dados tem sido usada com sucesso em domínios como ciências da saúde, indústrias químicas e computação científica, considerando que essas áreas exigem um mecanismo abrangente de rastreabilidade. Por outro lado, as empresas vêm aumentando a quantidade de dados que coletam de seus sistemas e processos, considerando a diminuição no custo das tecnologias de memória e armazenamento nos últimos anos. Assim, esta tese investiga se o uso de modelos e técnicas de proveniência é capaz de apoiar a análise da execução de processos de software e a tomada de decisões baseada em dados, considerando a disponibilização cada vez maior de dados relativos a processos pelas empresas. Um modelo de proveniência para processos de software foi desenvolvido e avaliado por especialistas em processos e proveniência, além de uma abordagem e ferramental de apoio para captura, armazenamento, inferência de novas informações e posterior análise e visualização dos dados de proveniência de processos. Um estudo de caso utilizando dados de processos da indústria foi conduzido para avaliação da abordagem e discussão de possibilidades distintas para análise e tomada de decisão orientada por estes dados.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

SUPPORTING SOFTWARE PROCESSES ANALYSIS AND DECISION-MAKING USING PROVENANCE DATA

Gabriella Castro Barbosa Costa Dalpra

October/2018

Advisors: Cláudia Maria Lima Werner
Regina Maria Maciel Braga Villela

Department: Systems Engineering and Computer Science

Data provenance can be defined as the description of the origins of a piece of data and the process by which it arrived in a database. Provenance has been successfully used in health sciences, chemical industries, and scientific computing, considering that these areas require a comprehensive traceability mechanism. Moreover, companies have been increasing the amount of data they collect from their systems and processes, considering the dropping cost of memory and storage technologies in the last years. Thus, this thesis investigates if the use of provenance models and techniques can support software processes execution analysis and data-driven decision-making, considering the increasing availability of process data provided by companies. A provenance model for software processes was developed and evaluated by experts in process and provenance area, in addition to an approach for capturing, storing, inferencing of implicit information, and visualization to software process provenance data. In addition, a case study using data from industry's processes was conducted to evaluate the approach, with a discussion about several specific analysis and data-driven decision-making possibilities.

SUMMARY

CHAPTER 1 – INTRODUCTION 1
1.1 Motivation 1
1.2 Problem, Hypothesis, and Research Questions 3
1.3 Goals 4
1.4 Research Methodology 5
1.5 Main Contributions 7
1.6 Text Structure 7
CHAPTER 2 – SOFTWARE PROCESS AND PROVENANCE DATA 9
2.1 Software Process 9
2.2 Provenance Data 12
2.3 Final Remarks 21
CHAPTER 3 – SYSTEMATIC MAPPING OF PROVENANCE IN THE CONTEXT OF SOFTWARE DEVELOPMENT PROCESSES 22
3.1 Planning Phase 23
3.2 Execution Phase 28
3.3 Reporting Phase 29
3.4 Review Findings and Discussion 39
3.5 Threats to Validity 42
3.6 Final Remarks 42
CHAPTER 4 – PROV-SwProcess PROVENANCE MODEL 44
4.1 Introduction 44
4.2 Relation with other standards / models 45
4.3 Aspects covered by PROV-SwProcess 46
4.4 PROV-SwProcess Model Specification 47
4.5 PROV-SwProcess Competency Questions 54
4.6 Final Remarks 61
CHAPTER 5 – iSPuP APPROACH 62
5.1 Introduction 62
5.2 iSPuP Phases 62
5.3 iSPuP Tool Support 64
5.4 iSPuP in Action 66

5.5 Final Remarks 81
CHAPTER 6 – PROV-SwProcess EVALUATION 83
6.1 Introduction 83
6.2 Materials and Method 83
6.3 Results and Discussion 86
6.4 Threats to Validity 90
6.5 Final Remarks 90
CHAPTER 7 – iSPuP EVALUATION 91
7.1 Introduction 91
7.2 Study Definition 91
7.3 Study Planning 92
7.4 Study Execution and Analysis 93
7.5 Results Discussion 112
7.6 Threats to Validity 116
7.7 Final Remarks 116
CHAPTER 8 – CONCLUSION 118
8.1 Epilogue 118
8.2 Contributions and Results 119
8.3 Open Questions and Future Work 121
REFERENCES 123
APPENDIX A - SELECTED STUDIES 132
APPENDIX B - STUDIES EXTRACTION AND QUALITY FORMS 133
APPENDIX C - PROV-SwProcess DISCREPANT CASES 147
APPENDIX D - SUBJECT CHARACTERIZATION FORM 152
APPENDIX E - EVALUATION FORM (VERSION 1) 154
APPENDIX F - EVALUATION FORM (VERSION 2) 162
APPENDIX G - INTERVIEW SCRIPT WITH COMPANY MANAGERS 171
APPENDIX H - SDP2: DETAILED EXECUTION AND ANALYSIS 182
APPENDIX I - SDP3: DETAILED EXECUTION AND ANALYSIS 193

LIST OF ILLUSTRATIONS

Figure 1.1: Research steps. 5

Figure 2.1: Nodes and edges in OPM (adapted from MOREAU et al., 2011). 14

Figure 2.2: Nodes and edges in PROV (adapted from GIL and MILES, 2013). 15

Figure 2.3: Process Provenance Data in Ontology. 18

Figure 2.4: wasAssociatedWith property chain (DALPRA et al., 2015). 19

Figure 2.5: Inference in Activity 'Solution_Implementation_-_2'. 21

Figure 3.1: Selection process results. 29

Figure 3.2: Number of papers x year. 30

Figure 3.3: Channel type. 31

Figure 3.4: Approaches evaluation. 38

Figure 3.5: Quality assessment results. 39

Figure 4.1: PROV-SwProcess - Retrospective Provenance (Part 1) 48

Figure 4.2: PROV-SwProcess - Retrospective Provenance (Part 2) 49

Figure 4.3: PROV-SwProcess - Prospective Provenance of Standard Process Level 49

Figure 4.4: PROV-SwProcess - Prospective Provenance of Intended Process Level. 50

Figure 4.5: PROV-SwProcess Inferences Example. 51

Figure 4.6: Example of provenance graph to support CQ1. 56

Figure 4.7: Tooltip when hovering the mouse on NULL Stakeholder. 56

Figure 5.1: Approach Execution Flow. 63

Figure 5.2: iSPuP Tool Architecture. 65

Figure 5.3: Process Manager Interface Example 1. 66

Figure 5.4: Process Manager Interface Example 2. 67

Figure 5.5: Data Table Example. 68

Figure 5.6: Toy Example Without Inferences. 70

Figure 5.7: Toy Example – Stakeholders Grouping Members. 70

Figure 5.8: Toy Example – Stakeholders Grouping Association. 71
Figure 5.9: Toy Example with Inferences. 71
Figure 5.10: Toy Example - Data Analysis Table. 72
Figure 5.11: Toy Example - Visualization to support CQ1. 73
Figure 5.12: Toy Example - Visualization to support CQ2. 74
Figure 5.13: Toy Example - Visualization to support CQ3. 74
Figure 5.14: Toy Example – Visualization to support CQ4. 75
Figure 5.15: Toy Example - Visualization to support CQ5 – part 1. 76
Figure 5.16: Toy Example - Visualization to support CQ5 – part 2. 77
Figure 5.17: Toy Example - Visualization to support CQ6. 77
Figure 5.18: Toy Example - Visualization to support CQ7. 78
Figure 5.19: Toy Example - Visualization to support CQ8. 79
Figure 5.20: Toy Example - Visualization to support CQ9. 80
Figure 5.21: Toy Example - Visualization to support CQ10 and CQ11 – part 1. 81
Figure 5.22: Toy Example - Visualization to support CQ10 and CQ11 – part 2. 81
Figure 6.1: Evaluation with Experts – First Round – Model Defects. 87
Figure 6.2: Evaluation with Experts – First Round – Defects Types. 87
Figure 7.1: SDP 1 – Flow Model with Activities and Roles. 94
Figure 7.2: SDP1 – Twenty-Five Instances Overview. 95
Figure 7.3: SDP1 - Stakeholders X Activities – Tabular View. 96
Figure 7.4: SDP1 – Activities Degree – Part 1. 97
Figure 7.5: SDP1 – Activities Degree – Part 2. 98
Figure 7.6: SDP1 - Stakeholders X Activities – Tabular View. 99
Figure 7.7: SDP1 - Stakeholders X Activities – Quality and DotNet activities. 99
Figure 7.8: SDP1 – All Stakeholders X Artifacts. 101
Figure 7.9: SDP1 - DotNet Associated Artifacts 101
Figure 7.10: SDP1 - Stakeholders X Created and Artifacts – Tabular View. 102

Figure 7.11: SDP1 – Artifacts Derivation. 104

Figure 7.12: SDP2 – Flow Model with Activities and Roles. 107

Figure 7.13: SDP2 – Ten Instances Overview. 108

Figure 7.14: SDP3 – Flow Model with Activities and Roles. 110

Figure 7.15: SDP3 – One Hundred and Thirty-Three Instances Overview..... 111

Figure 7.16: Results about the correctness of the analyses. 113

Figure 7.17: Results when checking if the performed analyses can assist in the proposed decision making. 114

Figure 7.18: Results when checking if the process manager can answer the CQs using his current process management tool or dashboard. 114

Figure 7.19: CQ Relevance. 115

Figure 7.20: CQ Relevance X Company. 115

Figure H.1: SDP2 - Tooltip. 183

Figure H.2: SDP2 – Activities Degree – Part 1. 184

Figure H.3: SDP2 – Activities Degree – Part 2. 185

Figure H.4: SDP2 - Stakeholders X Activities – Tabular View. 186

Figure H.5: SDP2 – All Stakeholders X Artifacts. 187

Figure H.6: SDP2 - Stakeholders X Created and Artifacts – Tabular View. 188

Figure H.7: SDP2 - Stakeholders x Roles. 190

Figure H.8: SDP2 – Artifacts Derivation. 192

Figure I.1: SDP3 – Null Stakeholder. 194

Figure I.2: SDP3 – Activities Degree – Part 1. 195

Figure I.3: SDP3 – Activities Degree – Part 2. 196

Figure I.4: SDP3 - Stakeholders X Activities – Tabular View. 197

Figure I.5: SDP3 – Stakeholders and Associated Artifacts. 198

Figure I.6: SDP2 - Stakeholders X Created and Artifacts – Tabular View. 199

Figure I.7: SDP2 - Stakeholders x Roles. 200

LIST OF TABLES

Table 2.1: Software Process Data. 20

Table 3.1: Research questions for the mapping (MQ) and review (RQ). 23

Table 3.2: Search string keywords. 25

Table 3.3: Quality assessment questionnaire. 27

Table 3.4: Extraction form. 28

Table 3.5: Authors' names and number of publications. 30

Table 3.6: Identified approaches. 32

Table 3.7: Identified provenance models. 34

Table 3.8: Approaches benefits. 35

Table 3.9: Provenance extraction. 36

CHAPTER 1 – INTRODUCTION

This chapter presents the motivation for the development of this thesis, the problem, hypothesis, and research questions that guided the approach proposal, as well as its goals and research methodology.

1.1 Motivation

Software applications and systems affect all business sectors and aspects of our daily life. Then, software development "*is a critical activity that needs to be carefully studied, understood, improved, and supported*" (FUGGETTA and DI NITTO, 2014).

Researchers and industry professionals have increasingly explored, since the 80's, techniques to improve software development processes (SDP) (HUMPHREY, 1989) and, nowadays, software development can still be considered a key activity to industry future growth, considering software as one of the most important industrial competitive factors (BOSCH, 2017).

Software process is "*a complex endeavor involving professionals, organizations, company policies, tools, and support environments*" (FUGGETTA and DI NITTO, 2014), and organizations have also invested on improving processes definition and management, based on the principle that the quality of software products is strongly related to the quality of the adopted processes to build them (FUGGETTA, 2000).

Due to the rapidly dropping cost of memory and storage technologies in the last years, companies have been dramatically increasing the amount of data that they collect from their systems (MCAFEE and BRYNJOLFSSON, 2012). During the software development, many different types of data can be generated and collected (DERNIAME *et al.*, 1999):

- **Product Data**: such as source code, configuration management data, documentation, executables, test suites, testing results, and simulations;
- **Process Data**: such as an explicit definition of a software process model, process enactment state information, data for process analysis and evolution, history data, project management data; and
- **Organizational Data**: such as ownership information for various project components, roles and responsibilities, and resource management data.

It is not a novelty that software development companies started to adopt data-driven practices in parts of their business over time (BIRD *et al*., 2011) (OLSSON and BOSCH, 2014). They have used data in accounting, marketing, and sales for calculating various performance indicators (such as return on investment for accounting, errors found in deployed products, and defect management). However, the use of software process data could be a challenging topic for many software engineers. Considering that engineering education "*tends to focus on formulas, clear cause effect relations and predictable behaviors of the systems built by engineers, the notion of statistical behavior, analysis of large data sets and the use of averages and deviations feels less tangible, or, if nothing else, requires an alternative mindset from the people working with the data*" (BOSCH, 2017). Buse and Zimmermann (2012) cite that there is "*a substantial disconnection between the information and insights needed by project managers to make good decisions and that which is typically available to them*". Bhattacharya (2012) affirms that "*the decision-making process in software development and maintenance is mostly dependent on software practitioner's experience and intuition*". Besides that, over time, the records accumulate, and the volume of data makes SDP data analysis even more difficult to be conducted.

One possible way to support software processes reproducibility and reduce the possibility of repeating failed executions is by using provenance data. For this end, it is important to store data both from the SDP and from the process execution. These data can be obtained using provenance techniques and models. Data provenance can be defined as the description of the origins of a piece of data and the process by which it arrived in a database (BUNEMAN *et al*., 2001). Tracking provenance enables sharing, discovering, and reusing the data, simplifying collaborative activities, reducing the possibility of repeating dead ends, and facilitating learning (RAM and LIU, 2007).

The importance of provenance has been widely recognized in the scientific workflow community (DAVIDSON *et al.*, 2007) (DAVIDSON and FREIRE, 2008) (ALAWINI *et al*., 2018). In this domain, provenance helps to interpret and understand the results, verify if the experiment was performed according to what has been defined and using acceptable procedures, identify the experiment's inputs, and reproduce the result (FREIRE *et al*., 2008). However, provenance has also been successfully used in other areas, mainly in complex domains, like health sciences, chemical industries, and scientific computing, taking into account that these areas require a comprehensive semantic traceability mechanism (BOSE and FREW, 2005 *apud* THAKUR *et al*.,

2009). The emergence of technologies such as Big Data, Cloud Computing, CyberSecurity, E-Science, and the increasing complexity of information systems made evident that traceability and provenance are promising approaches (LEAL *et al*., 2015). Considering that SDP is a complex domain and that the execution data should be controlled and evaluated to understand what really occurred during the execution of the process, the main idea of this thesis is to apply provenance techniques and models in the software processes domain, aiming to support SDP analysis and data-driven decision-making.

Gradually, the term provenance is being used in the context of SDP (XU and SENGUPTA, 2005). SDP stakeholders, such as developers, managers, and quality team members, want to understand "how and why a feature, component, chunk of code, test suite, or other development artifact came to be where it is". More than this, questions such as: "Where did this software product / entity (e.g., function, feature, test suite, documentation) come from?", "What is its history?", "What / and how other entities are related to it?", "Who else is using / used this?", "For what purpose was it generated?" or "How reliable is it?" are increasingly common in the SDP area and, using the provenance of the SDP data, we were able to correctly answer them.

In addition to answering the above questions it would be important to provide to the process manager, what occurred or was actually implemented, during the execution of the company's software processes, based on data from the process execution, aiming to support him / her in process decision-making.

1.2 Problem, Hypothesis, and Research Questions

Data and knowledge acquired in previous process executions can be reused to support a continuous process improvement. Since the 90's, capture and analysis are key elements in any strategy for software process improvement (WOLF and ROSENBLUM, 1993). Improving or designing new process requires to obtain concise, accurate and meaningful information about existing processes. After that, this information can be used to identify and eliminate problems and to develop and validate process improvements (WOLF and ROSENBLUM, 1993). Based on this and according to the motivation presented (Section 1.1), the main problem analyzed in this thesis is:

How to capture and analyze what really occurred during a software development process execution in order to support process analysis and data-driven decision-making?

Considering the use of data to confirm or disprove any beliefs and assumptions in an organization, this thesis hypothesis is:

The use of provenance models and techniques for capturing and analyzing software process provenance data can improve and assist process managers in the SDP analysis and support data-driven decision-making.

The presented hypothesis considers the existence of different systems for the SDP execution, the lack of a standard model to capture SDP provenance (considering the specificities of this domain when compared to processes in general), and the absence of an approach to support SDP provenance and execution data capture and storage, as well as the use of these data to support process managers in process analysis and decision-making activities. Based on these issues, the following research questions were formulated:

- ***RQ1***. *What SDP execution and provenance data should be captured?*
- ***RQ2***. *Which implicit information can be derived from captured data?*
- ***RQ3.*** *What are the characteristics and limitations of the existing provenance approaches / models that deal with SDP provenance?*
- ***RQ4.*** *What are the analysis possibilities that can be carried out on the captured data?*
- ***RQ5.*** *How SDP analysis can help in process manager decision-making?*

1.3 Goals

The main goal of this thesis is:

Develop and evaluate an approach for capturing, storing, discovering and visualizing SDP execution provenance data to support process analysis and data-driven decision-making.

This generic goal can be decomposed in the following specific goals:

1) Characterize existing works that use provenance in the context of SDP, by analyzing their features, strengths, and limitations;
2) Identify the necessary features for a model that aims to capture and query SDP provenance data;

3) Define a provenance model to deal with the specificities of SDP;
4) Define and implement an approach that captures, stores, analyzes and visualizes SDP provenance execution data showing what really occurred during the SDP execution, supporting process managers' process analysis and decision-making activities; and
5) Ensure that the proposed approach can support process managers in SDP analysis and decision-making activities, using real scenarios.

1.4 Research Methodology

This thesis was based on the Design Science Research (DSR) methodology. It seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts (HEVNER *et al.*, 2004). When the DSR methodology is used, "*the researcher learns about artifacts and natural settings by formulating hypotheses (a design), conducting an experiment (instantiating an artifact), and matching the results to the expectations (evaluating)"* (BASKERVILLE *et al*., 2009). Then, in order to answer the proposed research questions and check the research hypothesis, Figure 1.1 shows the main steps that were taken during the research.

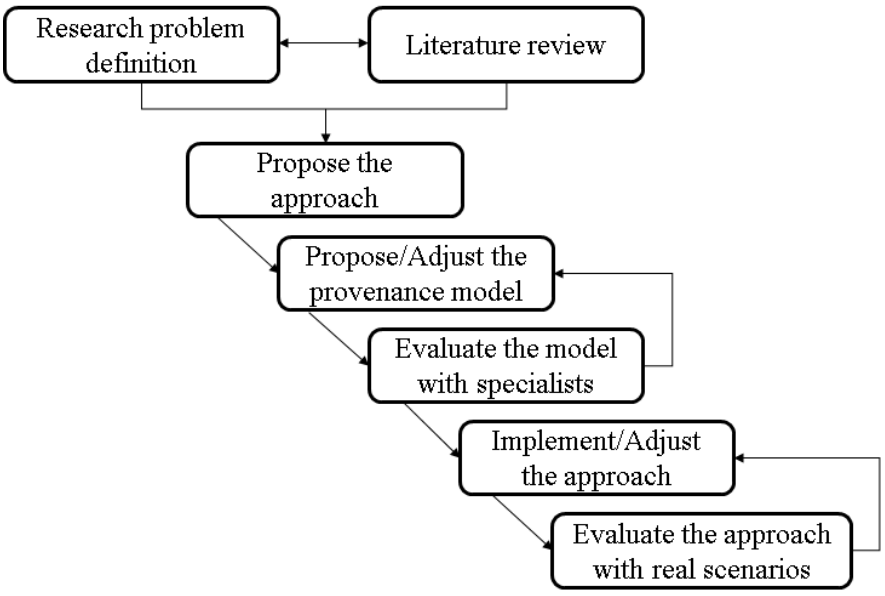


Figure 1.1: Research steps.

The first two steps (*Research problem definition and Literature review*) were performed iteratively. Initially, an informal literature review was done to get the initial and basic knowledge about the research topics and a gap considering the application of techniques and provenance models in the context of SDP was found. After that, the research problem was defined and a *quasi*-systematic review analyzing the use of

provenance in SDP was done, allowing a broader and more comprehensive vision about the thesis problem. Fourteen papers were selected to be analyzed. Provenance data began to be applied in the SDP domain after 2005 and there are few researchers or groups of researchers working in these two areas (provenance AND software processes). Only two authors appeared more than twice in the paper selection and with 3 publications and there are few approaches (35.7%) focused on making SDP provenance-aware.

In the third step (*Propose the approach*), an approach called iSPuP (improving Software Process using Provenance) was specified and some studies to evaluate its viability were performed (DALPRA *et al.*, 2015) (COSTA, 2016) (COSTA *et al.*, 2016a) (COSTA *et al.*, 2016b). The core of iSPuP approach is a provenance model called PROV-SwProcess. iSPuP supports PROV-SwProcess model instantiation, new information inferencing and data visualization.

PROV-SwProcess model was defined in the fourth step (*Propose the provenance model*). It was developed to accommodate SPD provenance specificities, including its main elements, relations, inference rules and competency questions. An evaluation about this model with provenance and process experts was conducted. It was planned as a model inspection and used a specific questionnaire to support the detection of possible semantic defects and improvements points in PROV-SwProcess. Two rounds of model evaluation were carried out, and three experts inspected PROV-SwProcess model. In the last round, the expert pointed out 32 correct points and 6 defects (3 incorrect facts, 1 inconsistence and 2 omissions). These defects were corrected in the model version presented in this thesis.

iSPuP approach and its tool support were implemented in the sixth step and, in the last step (*Evaluate the approach with real scenarios*), we *analyze* iSPuP approach and PROV-SwProcess provenance model to evaluate its feasibility *for the purpose of* supporting data analysis and data-driven decision making *with respect to* providing relevant information *under the point of view of* process managers *in the context of* software development process. Process data from three different companies and interviews with the process managers from these companies were used in a case study to evaluate the proposed approach. The case study showed that the use of the iSPuP approach, with PROV-SwProcess provenance model, is capable of assisting in making previously established decisions, and most of them would not be possible with the systems and tools currently adopted by the companies.

1.5 Main Contributions

This thesis has the following contributions:

- A *Quasi*-Systematic Literature Review of Provenance in the Context of Software Development Processes;
- PROV-SwProcess provenance model – a provenance model to SDP;
- A set of competence questions that can be answered using PROV-SwProcess and the respective decision-making possibilities that can be performed in answering these questions;
- iSPuP approach and its tool support to instantiate PROV-SwProcess model with process provenance data, new information inferencing, and data visualization (allowing data analysis and decision-making); and
- iSPuP evaluation, using three real scenarios with software process data execution.

1.6 Text Structure

The remaining of this text is organized as follows:

- Chapter 2 – Presents the main concepts related to software process and provenance data, including its main models proposed in literature.
- Chapter 3 - A *quasi*-systematic literature review and mapping, showing how provenance has been applied in the SDP domain by using a predefined methodology is presented in this chapter.
- Chapter 4 – This chapter presents PROV-SwProcess model, the provenance model developed to accommodate SP provenance specificities.
- Chapter 5 – An approach, called iSPuP, that supports PROV-SwProcess model instantiation, new information inferencing and data visualization, is detailed in this chapter, with its main elements and tool support.
- Chapter 6 – Details an evaluation with provenance and software process experts to inspect / validate PROV-SwProcess model.
- Chapter 7 – Presents the planning, execution, and results of an evaluation using real scenarios / process data from three different companies and interviews with the process managers from these companies in order to evaluate the main iSPuP elements.

- Chapter 8 – Conclusion summarizes the contributions of this thesis and presents the open questions and opportunities for the approach improvement.

CHAPTER 2 – SOFTWARE PROCESS AND PROVENANCE DATA

This chapter presents the main concepts used in this thesis, including software process and provenance data, including the main provenance models proposed in literature.

2.1 Software Process

Software process or software development process (SDP) is a critical factor for developing quality software products, considering it aims to manage and transform users' requirements into a software product that meets users' needs (ACUNA *et al*., 2000). SDP can be defined as:

- "A partially ordered set of activities undertaken to manage, develop and maintain software systems" (ACUNA *et al*., 2000);
- "A set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products" (PAULK, 2009).

In addition to the previous definitions, there are others in the literature (HUMPHREY, 1989) (LONCHAMP, 1993) (BENDRAOU and GERVAIS, 2007). An objective and complete definition of the software process that will be adopted in this work is "*the coherent set of policies, organizational structures, technologies, procedures, and artifacts that are needed to conceive, develop, deploy, and maintain a software product*" (FUGGETTA, 2000).

A well-defined SDP should indicate the activities to be executed, the required resources, produced and consumed artifacts, adopted procedures (methods, techniques, document models, etc.), and the criteria for carrying out the activities (BARRETO, 2011). The essential aspects of software development process considered in this thesis are: activities, stakeholder, resource, procedure, and artifact, as proposed by Falbo and Bertollo (2009). Each of these aspects is described in the following:

- **Activity***:* deals with the process activities used to create and/or maintain software and how they compose the software development process;
- **Stakeholder***:* refers to organizations, persons, projects, or teams acting or interested in the software process activities;

- **Resource***:* involves hardware equipment and software products used by the software process activities;
- **Procedure***:* relates to methods, techniques and document templates adopted by the software process activities; and
- **Artifact***:* represents different types of objects produced, changed, and used in process activities.

Another important concept about SDP is its life cycle. It covers the engineering activities of a process. The activities of this cycle are called meta-activities, and the life cycle is called software meta-process (DERNIAME *et al*., 1999). There are several life cycle proposals for software processes: DERNIAME *et al*. (1999) propose a life cycle called *PROMOTER Reference Model*; a reuse life cycle of software processes is presented by JØRGENSEN (2000) *apud* BARRETO (2011). NGUYEN and CONRADI (1994) identify a taxonomy to characterize meta-process categories, and their characteristics and compare some environments considering these categories and characteristics. REIS (2003) presents a detailed and more complete SDP life cycle, containing the following phases (or activities): technology provision, process requirements analysis, process design, process instantiation, process simulation, process execution and process evaluation. These phases are next detailed:

- **Technology Provision**: includes the technology provision to support the software and process model's production (such as process modeling languages, process models for reuse, tools for modeling, analysis, design, simulation, evolution, execution, and monitoring of software processes);
- **Process Requirement Analysis**: identifies the requirements for designing a new process or the new requirements for an existing process;
- **Process Design**: this phase can also be described as a process modeling step, which elicits and captures descriptions of informal processes, converting them into formal process models;
- **Process Instantiation**: in this phase, detailed information about the deadlines, agents and resources used by each activity defined in the process are added to the process model;
- **Process Simulation**: this phase allows the verification and validation of the defined process, before its execution;

- **Process Execution**: this phase uses the instantiated process and executes it invocating tools to guide and watch the execution of the modeled process. Information and metrics about the process progress can be collected and analyzed during this phase; and
- **Process Evaluation**: this phase aims to provide quantitative and qualitative information about the process execution performance; it can occur in parallel with the process execution and the acquired information can be used in the future occurrences of the process requirements analysis phase.

The approach and the provenance model presented in this thesis consider mainly the phases of execution and analysis of the presented software processes life cycle. During the execution phase, SDP data are *captured* in order to be *analyzed* during the process evaluation.

Process analysis can be of two different types (WOLF and ROSENBLUM, 1993):

- **Deductive Analysis**: is concerned with analyzing an abstract specification of a process in some formal logic, with the goal of discovering inconsistencies or other anomalies that would be present in enactments of the process; and
- **Retrospective Analysis**: is concerned with analyzing empirically gathered data from several enactments of a process, with the goal of discovering patterns of anomalous behavior that can be eliminated in future enactments.

This thesis approach deals with *retrospective analysis* and PROV-SwProcess model was developed to provide the fundamental information required to understand and analyze SDP provenance data. This model defines SDP constructs (activities, stakeholder, resource, procedure, and artifact) and several causal relations that can happen between these constructs during the process execution (e.g., created and modified artifacts, procedures adoption, activities and stakeholders' associations, etc.). These relations represent some cause-and-effect influences that can be established between SDP data, allowing a deeper understanding and interpretation of SDP execution.

The proposed approach also considers the storage and inference of new information from the capture of both *prospective* and *retrospective* provenance of software processes (the definition of these terms is presented in the next section).

2.2 Provenance Data

Data provenance can be defined as the origins description of a piece of data and their processing history (BUNEMAN *et al*., 2001) or a record of the data derivation history, which enables reproducibility, interpretation of results and diagnosis of problems (LIM *et al*., 2010). According to Herschel *et al*., (2017), provenance can be seen as meta-data that, instead of describing data, describes a production process. It brings transparency and help to audit and interpret data (MOREAU, 2010) (CUEVAS-VICENTTÍN *et al*., 2016). Capturing and processing of provenance are important in various settings, e.g., to assess quality, to ensure reproducibility, or to reinforce trust in the end product (HERSCHEL *et al*., 2017).

Provenance data differs from traditional data items and meta-data considering that it is an immutable directed graph, incrementally captured at run-time (SUN *et al*., 2013). Nevertheless, the capture of process provenance data does not interfere in the process execution and allows the process managers or other process data analysts to refine the applied filtering rules for data process collection (GHOSHAL and PLALE, 2013).

In addition to being related to the data, the term provenance may also be associated with the process(es) that enabled the data creation (SIMMHAN *et al*., 2005) (CRUZ *et al*., 2009).

According to Freire *et al.* (2008), provenance from computational tasks can be divided into two types: (i) *prospective provenance* that captures a computational task's specification and corresponds to the steps that must be followed to generate a data product, and (ii) *retrospective provenance* that captures the steps executed as well as information about the environment used to derive a specific data product.

The capture and use of provenance data in the context of software processes can be specified into the process lifecycle as follows:

- in the Process Design (or Process Modeling) and Process Instantiation phases, the created models must be properly stored, in order to allow the capture of the prospective provenance (provenance related to the process specification);
- if previously captured data already exist, they can be used in the Process Simulation phase, supporting process verification and validation, before process execution in real scenarios;

- during the Process Execution phase, SDP provenance data (retrospective provenance) can be captured (provenance models can be used to provide a standard model for capturing these data) and the data must be appropriately handled to be used during the Process Evaluation phase; and
- in the Process Evaluation phase, all the captured/stored provenance data can be used to derive information that may contribute to the improvement of the process initially defined and to assist the process manager in making some specific strategic decision.

To obtain the benefits of provenance information, provenance data should be captured/stored in an integrated manner to allow queries on that data. In this vein, there are two main models proposed in the literature: OPM (MOREAU *et al.*, 2011) and PROV[1] (MOREAU and GROTH, 2013), which are cited or used by some works analyzed in the review presented in Chapter 3 (SUN *et al.*, 2013) (WENDEL *et al.*, 2010) (COSTA *et al.*, 2016b) (GODFREY, 2015) (COSTA, 2016) (DALPRA *et al.*, 2015). These two models, as well as a new model proposed by a master's thesis developed in the context of this thesis (called PROV-Process), are presented in the following.

2.2.1 OPM

The OPM - Open Provenance Model (MOREAU *et al.*, 2011) was created in order to allow the provenance metadata interoperability between different systems. This model was designed to meet the following main requirements:

- Allow provenance information to be exchanged between systems through a compatibility layer, based on a shared provenance model;
- Allow developers to create tools that operate on such provenance model;
- Define the model in a formal way;
- Support a digital representation of provenance for any "thing", produced or not, by computer systems; and
- Define a set of rules that identify valid inferences and that can generate graphs of provenance.

OPM uses a graph to represent the provenance information. In this graph, there are nodes or vertices called artifacts (A), processes (P), and agents (Ag); and its edges, as

[1] An overview of PROV model can be found in https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

shown in Figure 2.1. **Artifacts** are immutable pieces of state, which may have a physical embodiment in a physical object (or a digital representation in a computer system). **Processes** represent an action or series of actions performed on (or caused by) artifacts, resulting in new artifacts, and **agents** are entities acting as a catalyst of a process, enabling, facilitating, controlling, or affecting process execution (MOREAU *et al.*, 2011).

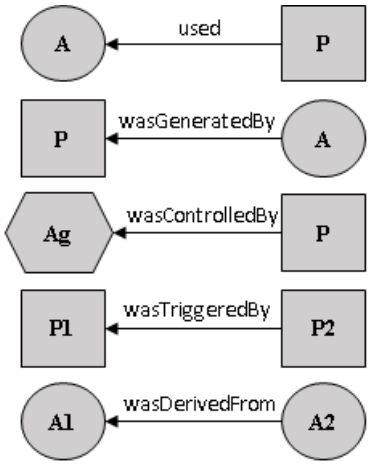


Figure 2.1: Nodes and edges in OPM (adapted from MOREAU *et al.*, 2011).

In order to capture the dependencies between the artifacts, processes and agents, an edge is used. It represents the causal dependence between its source (denoting the effect) and its destination (denoting the cause). As shown in Figure 2.1, OPM has 5 causal dependencies: **used**, **wasGeneratedBy**, **wasControlledBy**, **wasTriggeredBy** and **wasDerivedFrom**. Three of them (**used**, **wasGeneratedBy**, and **wasControlledBy**) can be associated with a role. Roles are used to distinguish the nature of the dependency when multiple edges are connected to the same process. As examples of roles, proposed by Moreau *et al*. (2011), we can cite: "a gardener may control the digging process (role = "dig the bed"), as well as planting a rose bush (role = "plant") and watering the bush (role = "irrigating")".

A more generic provenance model, called PROV, was specified by W3C working group (MOREAU and GROTH, 2013). In addition to the basic characteristics of OPM, PROV model presents new constructions and relations. Two nodes in OPM were changed and new causal relationships were created. This model is detailed in the next subsection.

2.2.2 PROV

PROV model (MOREAU and GROTH, 2013) aims to express provenance data through the description of entities, activities, and agents (all represented by vertices)

involved in producing or delivering an object and the causal relationships between them (represented by edges). The goal of PROV provenance model is to enable the publication and interchange of provenance information in heterogeneous environments. PROV differs from OPM by the name of two vertices (entity, activity) and presents new causal relationships[2], as can be seen in Figure 2.2 (this figure shows only the main relationships of PROV; it should be emphasized, however, that the model offers others, which are derived from these seven main ones).

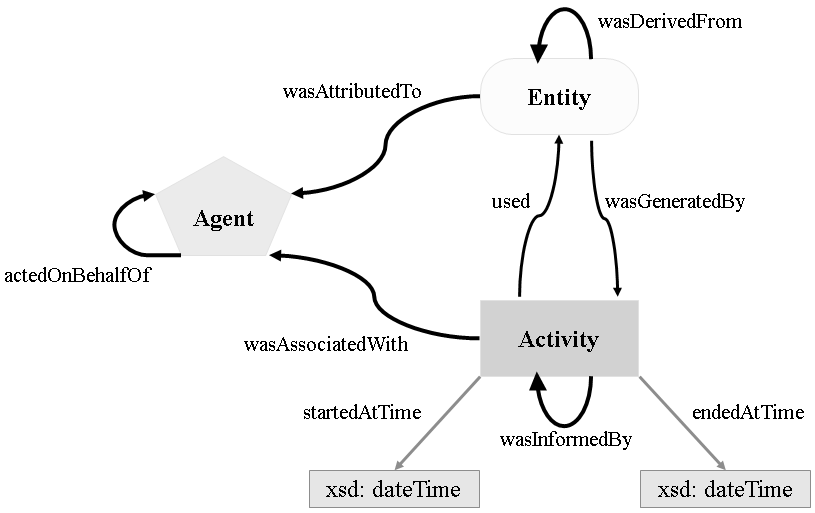


Figure 2.2: Nodes and edges in PROV (adapted from GIL and MILES, 2013).

The main PROV causal relationships are presented next:

- **used**: lists activities, stating that one activity used an entity;
- **wasGeneratedBy**: relates entities to activities and indicates that an entity was generated by an activity;
- **wasAssociatedWith**: relates activities and agents, indicating that an activity has been associated with an agent;
- **wasAttributedTo**: relates entities and agents and indicates that an entity has been assigned to an agent;
- **actedOnBehalfOf**: lists agents indicating that an agent has authority or responsibility for another agent;
- **wasDerivedFrom**: relates entities, in the sense that one entity originated from the other. This derivation has the evolutionary character, and not corrective; and

[2] A detailed description of PROV and its relations can be found in https://www.w3.org/TR/2013/REC-prov-dm-20130430/.

- **wasInformedBy**: relates activities implying that an informed activity has been generated by the activity that reported it, but this activity is unknown or not of interest.

PROV has a family of documents (GROTH and MOREAU, 2013) that defines a model, corresponding to serializations and other supporting definitions to enable the interoperable interchange of provenance information in heterogeneous environments. This family has as core a conceptual data model (PROV-DM), which defines a common vocabulary used to describe provenance. Besides that, PROV family has other three recommendations: (i) PROV-O: PROV ontology, an OWL2 ontology allowing the mapping of PROV data model to RDF; (ii) PROV-N: a notation for provenance aimed at human consumption; and (iii) PROV-CONSTRAINTS: a set of constraints applied to PROV data model.

Considering PROV model is generic and presents several possibilities of causal relationships, there are in the literature some proposals to specialize this model to specific domains, such as D-PROV (MISSIER *et al*., 2013b), ProvONE (CUEVAS-VICENTTÍN *et al*., 2016) and Versioned-PROV (PIMENTEL *et al*., 2018).

D-PROV (MISSIER *et al*., 2013b) extends the PROV model to represent the process structure, i.e., to enable prospective provenance storing and querying. D-PROV was a previous incarnation of ProvONE (CUEVAS-VICENTTÍN *et al*., 2016).

ProvONE is a model for scientific workflow provenance that extends PROV with its specific structure elements. ProvONE was developed in the context of DataONE Project (DATA ONE, 2018), a large scale and federated data infrastructure for the earth sciences community. Although this model is useful in scientific workflow domain, it does not suffice for capturing and analyzing provenance in the software development process domain. For example, in ProvONE, the workflow execution corresponds to the execution of computational tasks *only* by software agents but, in the software process context, we need to express different types of agents, such as, person, software agent and organizations. Besides, ProvONE does not propose new rules to derive implicit provenance information. Taking into account the gaps of ProvONE and that PROV does not capture the specificities of software development processes, extensions in this model should be made. An initial effort in this context was made by a master's thesis developed in the context of this thesis (called PROV-Process) and is presented in the following subsection.

Another PROV extension is Versioned-PROV (PIMENTEL *et al*., 2018). It adds support for the provenance of mutable values by time-versioning entities, being useful to represent fine-grained provenance from scripts. Versioned-PROV considers that "*PROV does not properly support fine-grained provenance with mutable data structures due the assumption of immutable entities and their representation may become quite verbose*".

2.2.3 PROV-Process

PROV-Process approach (DALPRA *et al*., 2015) (COSTA, 2016) is an architecture for capturing, storing, and analyzing processes provenance data, using PROV. This architecture uses a data model, extended from PROV-DM[3] specifications (MOREAU and MISSIER, 2013), to capture and store software process provenance data properly. After the software process provenance data are captured and stored in this database, semantic web technologies (ontologies and inference machines) can be applied to derive implicit information that can be useful to the process manager for analyzing the process data.

PROV-Process approach offers a web system[4] with a relational database based on PROV-DM and permits to build a computational ontology, specified using the Ontology Web Language (OWL). This ontology, named PROV-Process Ontology[5], is an extension of PROV-O (LEBO *et al*., 2013) and includes all the provenance data captured from the software development process. While a reference ontology defines a formal and explicit specification of a shared conceptualization and allows capturing the common understanding of objects and their relationships in a given domain (GUARINO, 1998), a computational (or operational) ontology is obtained from a reference ontology. PROV-O (LEBO *et al*., 2013) represents the PROV Data Model using OWL. It provides a set of classes, properties, and restrictions to represent provenance information. Furthermore, OWL is based on logic specification, then, it is possible to use inference mechanisms in this language. With this mechanism we can derive new information and relationships that were previously implicit.

When exported to the ontology, process provenance data is transformed into ontology individuals. These individuals were represented in the PROV-Process database

[3]PROV-DM is the conceptual data model from PROV.

[4] The developed system to support the PROV-Process approach is available in https://github.com/humbertodalpra/ProvProcess

[5] Available at https://goo.gl/zBDNfc

as records in the tables Activity, Entity, Agent, and all the other tables used to store the relationships between the software process provenance data. An illustrative example of how software process provenance data were imported in PROV-Process Ontology can be seen in Figure 2.3. This figure presents data using an open-source system to develop and maintain ontologies, called Protégé[6], with a PROV-Process Ontology example. As shown in this figure, there is the task *Opening_the_Request_for_Change_1* and it is an individual of the *Activity* class. This task is associated with the actor *Client_1* using the object property *wasAssociatedWith* (a PROV relationship) and has some related data properties, such as its *start time* and *process instance id*.

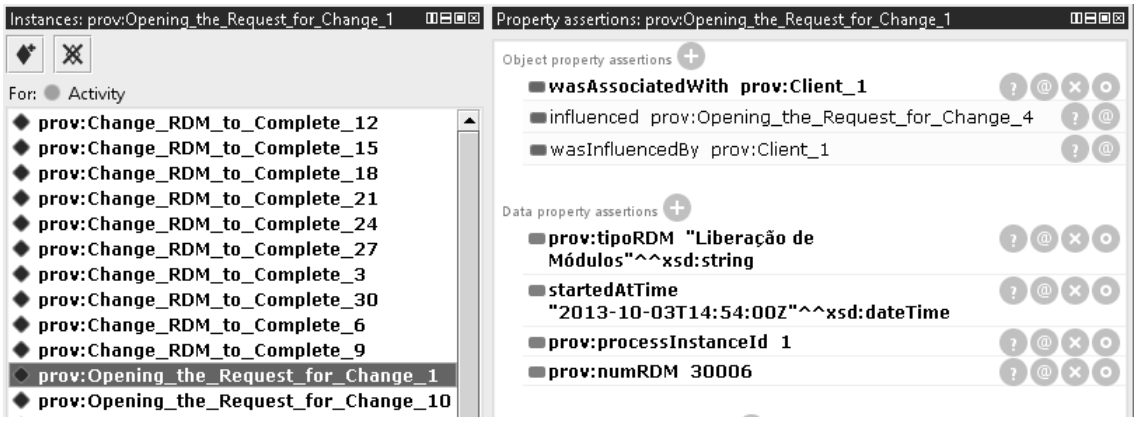


Figure 2.3: Process Provenance Data in Ontology.

One of the main advantages of the PROV-Process architecture is the possibility to make inferences about the obtained process data. The inferences were possible based on a group of rules (using Property Chains[7]) from PROV-Process Ontology. These properties chains allow the inference of implicit information about the software processes. In PROV-Process, three specific rules were added as sub properties in the 'wasAssociatedWith' data property. As shown in Figure 2.4, these rules state that if an activity *used*, *was started by*, or *was ended by* an entity and that entity was assigned to an agent, we can infer that an activity is associated with an agent. The formal specifications of these rules using OWL are:

- *used* **o** *wasAttributedTo* **SubPropertyOf** *wasAssociatedWith*
- *wasStartedBy* **o** *wasAttributedTo* **SubPropertyOf** *wasAssociatedWith*

[6] http://protege.stanford.edu/

[7] A property chain is a property that is defined as a series of other properties (W3C, 2012). Considering that property chains are formed by the connection of other properties, the domain of the first property in the chain must be the same domain of the property that is being formed. The domain of a property that is connected to another must be the same range as the class of the property that precedes it in the chain, and the last property in the chain must have the same class range as the range of the property being created.

- wasEndedBy **o** wasAttributedTo **SubPropertyOf** wasAssociatedWith

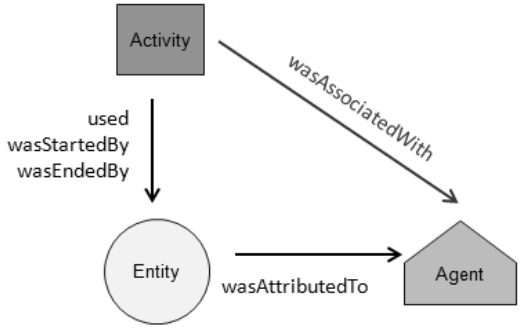


Figure 2.4: wasAssociatedWith property chain (DALPRA *et al.*, 2015).

An example of how this inference mechanism can be useful for the analysis of the process execution data is presented below, using data from Table 2.1. These data were obtained from a process that deals with changes in a software product. In this process, a change request in a software is opened and several people work on implementing this solution until its closure / solution.

After the execution of the inference machine on the PROV-Process Ontology, populated with the software process data[8] (Table 2.1), relationships that were not explicit in the execution data were derived. Examples of these relationships are two implicit relations about the activity *Solution_Implementation_-_2* (Figure 2.5[9]): (1) *wasAssociatedWith* `favio.riviera`, and (2) *wasAssociatedWith* `helen.kelly`. While in the process execution data it was explicit that during the execution of the *Solution Implementation* (id=2) only the actor `april.sanchez` was involved in this task, this inference brings new information. It states that the actors `favio.riveira` and `helen.kelly` could be involved in the execution of this task, considering that during the execution of other tasks of the process, these actors manipulated the same artifact (*DLL - Calculation*) that was used in the task *Solution Implementation* (id=2). Thus, in a next execution of this same process (if it will be done by the same team), the project manager, with this information, could suggest that the three actors work together on the task *Solution Implementation*, which could avoid many repetitions of the *Implementation* task until the requested change in the system is approved.

During the development of this thesis, PROV-Process was revised, new constructors were added (*Software_Process*, *Procedure*, *Resouce*, and its respective subtypes) or renamed (*Entities* are called *Artifacts* and five specific artifacts subtypes

[8] This ontology with its individuals is available at: https://goo.gl/evXcbr

[9] When using Protégé, the results returned by the inference engine can be visualized in the rectangles with beige background color.

were included), new relations between these constructs were specified and eight groups of inference rules were carefully defined and implemented, allowing the derivation of implicit information. Considering that many inclusions were made in the previous model, it was renamed PROV-SwProcess, and it is presented in detail in Chapter 4.

Table 2.1: Software Process Data.

ID	Task	Start Time	End Time	Artifacts	Actor
1	Open Change Request	2015-05-04 09:00:00	2015-05-04 09:05:00	-	marc.marseau
2	Solution Implementation	2015-05-04 09:10:00	2015-05-04 10:30:00	DLL - Calculation	april.sanchez
3	Solution Implementation	2015-05-04 11:00:00	2015-05-04 14:00:00	DLL - Calculation	helen.kelly
4	Solution Implementation	2015-05-04 16:00:00	2015-05-04 16:35:00	DLL - Calculation DLL - NF-e v2.0	helen.kelly, favio.riviera
5	Solution Implementation	2015-05-05 09:00:00	2015-05-05 12:00:00	DLL - ERP	anthony.nichols
6	Solution Implementation	2015-05-05 14:35:00	2015-05-05 15:05:00	DLL - Calculation DLL - ERP	favio.riviera
7	Solution Implementation	2015-05-06 09:10:00	2015-05-06 11:05:00	DLL - ERP	april.sanchez
8	Close Change Request	2015-05-06 14:00:00	2015-05-06 14:35:00	-	marc.marseau

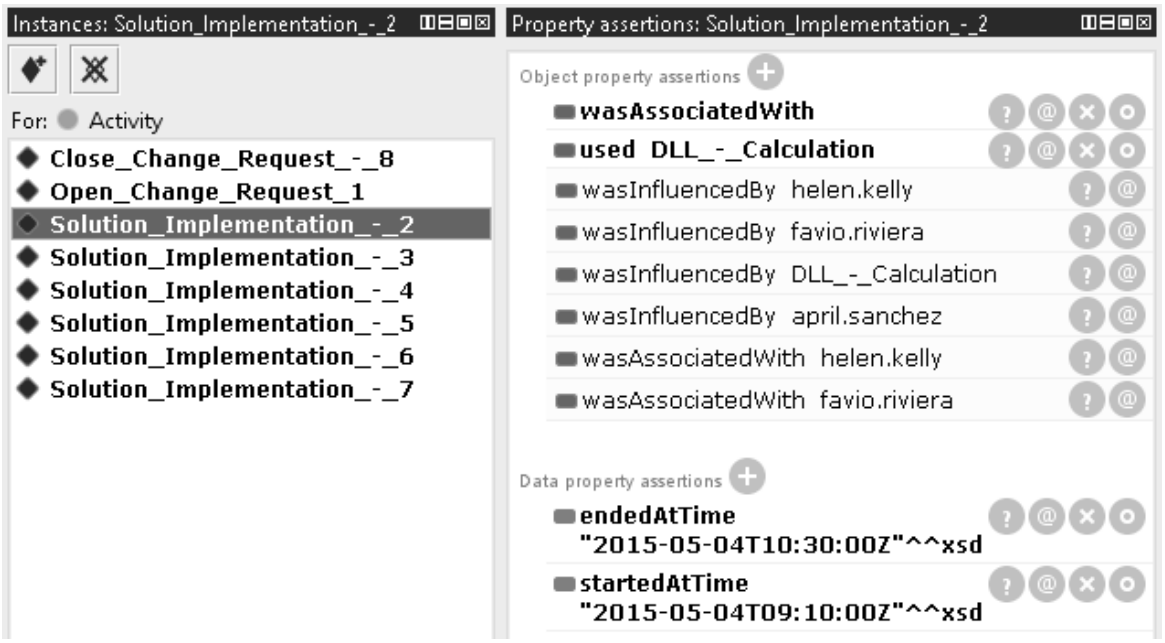


Figure 2.5: Inference in Activity 'Solution_Implementation_-_2'.

2.3 Final Remarks

This chapter presented the main concepts involved in this thesis, i.e., software processes and provenance data. SDP definition, their main components and a software development process life cycle were presented in Section 2.1.

In the section about provenance data (Section 2.2), this concept was explained as well as the main models presented in the literature (OPM and PROV), besides the PROV-Process model, which originated this thesis model (Chapter 4).

CHAPTER 3 – SYSTEMATIC MAPPING OF PROVENANCE IN THE CONTEXT OF SOFTWARE DEVELOPMENT PROCESSES

This chapter presents a quasi-systematic literature review and mapping, showing how provenance has been applied in the SDP domain by using a predefined methodology.

In order to evaluate the use of provenance in the context of software development process, a *quasi*-systematic literature review was planned and executed. A systematic literature review aims to evaluate and interpret all available research relevant to a specific research question, topic area, or phenomenon of interest (KITCHENHAM and CHARTERS, 2007). In the context of this thesis, we want to make a characterization review, investigating how provenance has been applied in the SDP domain by using a trustworthy and auditable methodology. Then, a *Quasi-Systematic Review* (TRAVASSOS *et al*. 2008) was planned and performed, considering this study must explore the same rigor and formalism for the methodological phases of protocol preparation and running of a systematic literature review, without performing any kind of comparison. This sort of review can also be identified as a *systematic mapping* (KITCHENHAM and CHARTERS, 2007).

Our research method uses the guidelines provided in Brereton *et al*. (2007), and consists of the following phases, with their respective activities:

- **Phase 1: Plan Review -** During this phase, the researcher specifies the review objectives and research questions, and develops the review protocol. After that, the protocol must be validated (before being applied), establishing its feasibility.
    - *1 - Specify Research Questions*
    - *2 - Develop Review Protocol*
    - *3 - Validate Review Protocol*
- **Phase 2: Conduct Review -** In this phase, the search strings are performed in the defined digital libraries and the obtained studies are evaluated according to the protocol criteria. Relevant data from the selected papers are extracted / synthesized.
    - *4 - Identify Relevant Research*
    - *5 - Select Primary Studies*

6 - Assess Study Quality

7 - Extract Required Data

8 - Synthesize Data

- **Phase 3: Document Review:** During this phase, the results of the systematic review are reported and validated.

  9 - Write Review Report

  10 - Validate Report

All these phases are presented in detail in the next sections.

3.1 Planning Phase

In the *Planning Phase*, the research questions and a review protocol must be defined, as presented in the following.

The overall objective of our literature review is to identify approaches that use provenance techniques and/or models in the context of SDP to obtain a more detailed and comprehensive view on this topic. Based on this objective, the mapping and research questions presented in Table 3.1 were developed.

Table 3.1: Research questions for the mapping (MQ) and review (RQ).

ID	Question
MQ1	How many studies were published over the years?
MQ2	Who are the most active authors in the area?
MQ3	Which publication vehicles are the main targets for research production in the area?
RQ1	What are the approaches that apply provenance in SDP domain?
RQ2	What are the provenance models for applying provenance in SDP domain?
RQ3	What are the benefits that can be achieved by using the approach?
RQ4	How was SDP provenance data extracted, stored, and analyzed?
RQ5	How was the approach evaluated?

Considering the research questions, the review scope was defined based on the PICO approach (PAI *et al*., 2004). This approach separates the question into Population of interest, Intervention or exposure being evaluated, Comparison intervention (if applicable) and Outcome. As this review aims mainly at characterizing the state-of-the-art, no comparison is carried out, i.e., it can be classified as a *quasi*-systematic review (TRAVASSOS *et al*. 2008).

- *Population* (P): Software development process.
- *Intervention* (I): Provenance data.
- *Comparison* (C): Not applied.
- *Outcomes* (O): Approaches.

In order to provide an initial understanding about the use of provenance in the context of software development processes, to assist in the search keywords definition, and to test / calibrate the search string, the following control papers were defined. These three control papers were obtained during a previous *ad hoc* literature review.

1. DANG, Y. B., CHENG, P., LUO, L., CHO, A. A code provenance management tool for IP-aware software development. In: Companion of the 30th International Conference on Software Engineering, Informal Research Demonstrations. ACM. pp. 975-976, 2008.
2. WENDEL, H., KUNDE, M., SCHREIBER, A. Provenance of software development processes. In: McGuinness D.L., Michaelis J.R., Moreau L. (eds) Provenance and Annotation of Data and Processes. IPAW 2010. Lecture Notes in Computer Science, vol 6378. Springer, Berlin, Heidelberg, pp.59-63, 2010.
3. COSTA, G. C. B., WERNER, C. M., BRAGA, R. Software Process Performance Improvement Using Data Provenance and Ontology. In: International Conference on Business Process Management. Springer International Publishing, pp. 55-71, 2016.

The search was done in three electronic databases:

- *Compendex* (www.engineeringvillage.com)
- *IEEExplore* (www.ieeexplore.ieee.org)
- *Scopus* (www.scopus.com)

These databases were chosen according to the following criteria (COSTA and MURTA, 2013 *apud* NEIVA *et al*., 2016):

- They allow using logical expressions or a similar mechanism;
- They allow full-length searches;
- They are available in the researcher's institution; and
- They cover the review research area: computer science.

ACM Library and ScienceDirect were not included among the selected sources because the ACM Library has its content indexed by the Scopus library and ScienceDirect did not bring any result using the keywords presented.

Although the IEEExplore database had not returned any of the control papers, it was used in this review because, by executing the search string, it returned some articles considered, initially, relevant. The non-return of the control papers by this database is justified by the fact that they are not indexed by it.

In order to establish the search string, we considered the terms presented in the PICO structure, its alternative spellings and synonyms, as listed in Table 3.2.

Table 3.2: Search string keywords.

	Category	Keywords
P	software development process	software process, software processes, software development, system development, systems development
I	provenance data	provenance
C	-	-
O	Approaches	approach, technique, method, methodology, tool, system, application, proposal

Considering the terms in Table 3.2, our string was structured using them and boolean OR/AND operators. Synonyms and alternate spellings were concatenated using OR and, after that, the terms of each PICO category were concatenated using AND. The final search string was:

*(“software process” **OR** “software processes” **OR** “software development” **OR** “system development” **OR** “systems development”) **AND** (“provenance”) **AND** (“approach” **OR** “technique” **OR** “method” **OR** “methodology” **OR** “tool” **OR** “system” **OR** “application” or “proposal”)*

The review includes every paper returned by the search string that meets at least one of the following inclusion criteria (IC) and does not meet any option of the exclusion criteria (EC):

- IC1 - Publications must address the use of provenance in the context of software process;
- IC2 - Publications must discuss opportunities and challenges by applying provenance in the context of software process;
- IC3 - Publications must present proposals and/or models for applying provenance in the context of software process; and

- IC4 - Publications must report experiences about the use of provenance in the context of software process.

The following exclusion criteria were established:

- EC1 - Publications not written in English;
- EC2 - Publications whose full text is not available for download, in their complete form, in the digital libraries, nor through any other way without costs for the researcher;
- EC3 - Publications not published in conferences, journals, workshops or seminars;EC4 - Publications not addressing software process AND provenance; and
- EC5 - If the same study has been published more than once, the most detailed version will be used (the others will be excluded).

The process to select the relevant publications for this review has seven steps:

1. **Search string execution**: in this step, the search string was executed in the data sources previously presented and the obtained results were cataloged for further analysis;
2. **Results merging**: the results from all databases were merged in JabRef (JABREF, 2017);
3. **1st filter - Remove duplicates**: using JabRef, duplicated results were removed;
4. **2nd filter - Analyze titles and abstracts**: the results were analyzed based on their titles and abstracts, considering the inclusion/exclusion criteria. The results clearly considered irrelevant were excluded. To reduce the risk of excluding a result at an early stage of the review, two doctoral students evaluated if the result should be included/excluded. If one suggested the inclusion and the other student suggested its exclusion, we chose to include the result.
5. **3nd filter - Analyze full text:** the results selected in the previous step were fully read and verified if the paper should be included and analyzed.
6. **Study data extraction** considering the research questions. In addition to data extraction, the articles selected on the 5th step were also evaluated using a quality assessment form.
7. **Snowballing** - a snowballing process was also performed using the papers selected in the 5th step. Their references were reviewed to find other potential primary studies.

The studies quality assessment cited in Step 6 can be used to (KITCHENHAM and CHARTERS, 2007):

- Provide even more detailed inclusion/exclusion criteria;
- Investigate whether quality differences provide an explanation for differences in study results;
- As a means of weighting the importance of individual studies when results are being synthesized;
- Guide the interpretation of findings and determine the strength of inferences; and
- Guide recommendations for further research.

In this review, the quality assessment was used to guide the interpretation of findings and determine the strength of inferences. However, it must be emphasized that the quality assessment performed is a judgment about what was reported in the publications rather than on the study quality. In this way, a customized quality assessment checklist was developed, based on the checklist suggestion provided in (KITCHENHAM and CHARTERS, 2007).

The questions in Table 3.3 were answered for each publication after the data extraction process. Each one of the six questions could score 1 point if the answer was "Yes", 0.5 point if the answer was "Partial" or 0 point if the answer was "No". According to this score, each publication could obtain a score from 0 to 6 points.

Table 3.3: Quality assessment questionnaire.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes/No/Partial
QA2	Is the presented approach clearly explained?	Yes/No/Partial
QA3	Is the used provenance model clearly described and their adoption justified?	Yes/No/Partial
QA4	Is there any empirical/experimental result regarding the approach?	Yes/No/Partial
QA5	Are threats to validity taken into consideration?	Yes/No/Partial
QA6	Are all research questions answered adequately?	Yes/No/Partial

A data extraction form was designed to gather the information necessary to answer the research questions (Table 3.4). Thus, for each publication approved by the selection process (see Appendix A), this information was extracted after reading all the papers.

Table 3.4: Extraction form.

Information Type	Data
Reference Information:	Title of document: Author(s): Publication date: Source:
Information from RQ1:	Approach name: Approach description:
Information from RQ2:	Provenance model name or description (if a model created by the authors is used):
Information from RQ3:	List of approach benefits:
Information from RQ4:	Artifacts whose provenance was extracted: Form of provenance data storage: Provenance Analysis method:
Information from RQ5:	Evaluation description:

3.2 Execution Phase

After the review planning (Phase 1), the review protocol can be applied. This review was first carried out during July 2017 and revised on June 2018. The obtained results in each of the steps defined in the review protocol are explained. Figure 3.1 shows the results obtained in each step of the Selection Process. Initially, the execution of the protocol in the search engines returned a total of 112 publications until 2017. From 2017 up to 2018, 13 publications were returned. These papers were merged and submitted to a filtering process, comprising three steps (Steps 3 to Step 5). Steps 2 and 3 were supported by JabRef (JABREF, 2017). Steps 4 and 5 were done manually, by two doctoral students, and Steps 6 and 7 were done by one and revised by the other. All 14 results selected in Step 5 are listed in Appendix A and all the information extracted from these studies can be found in Appendix B. In Step 7, all the references of these 14 studies were examined, and other 3 studies were selected to be analyzed in this review: Miles (2010), Munroe *et al*. (2006) and Miles *et al*. (2011). Considering that the last two papers mentioned describe the same approach (PrIMe) and are from the same group of authors, Miles *et al*. (2011) was considered for analysis as it is the most recent one about the proposed approach. However, after evaluating these approaches[10] according to the exclusion criteria, they fit into EC4: *Publications not addressing software process*

[10] Miles (2010) proposes the approach SourceSource, that adapts source code from its original form to record information on data provenance during execution, without manual manipulation, and Miles et al. (2011) present PrIMe, a software engineering technique for adapting application designs to enable them to interact with a provenance middleware layer, thereby making them provenance-aware.

AND provenance. Even dealing with provenance data and software development, they do not specifically address software processes.

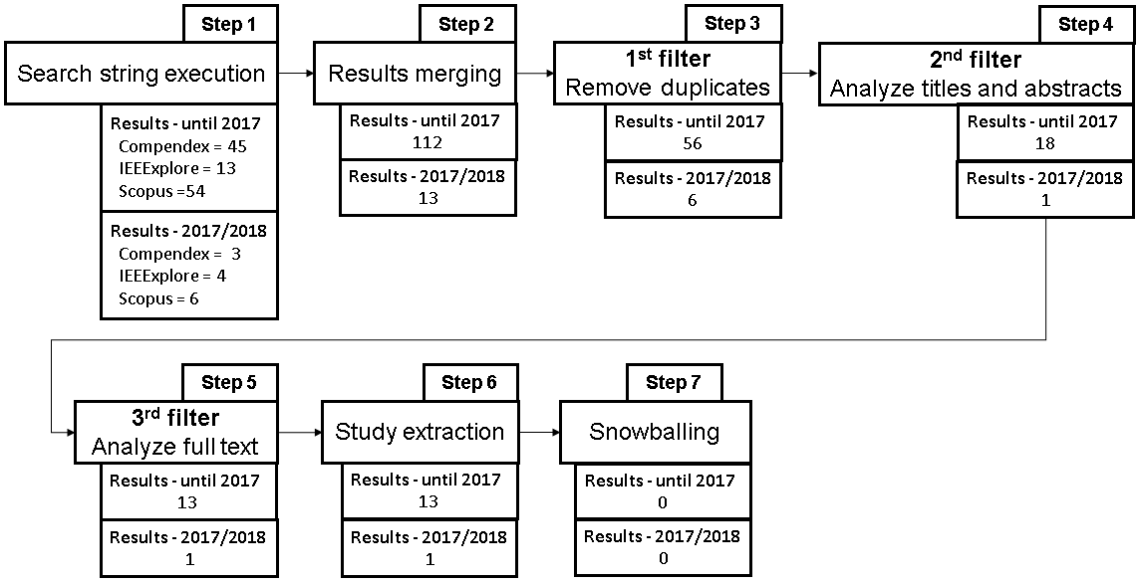


Figure 3.1: Selection process results.

3.3 Reporting Phase

This section reports and discusses the results obtained by the quasi-systematic mapping and review. 14 papers were actually selected as the result of the review execution. This means that 22.58% of the documents initially obtained (after removing the duplicates) actually contributed to this review. All other papers were excluded due to lack of any direct contribution to characterizing the use of provenance in SDP context.

- **Systematic mapping report**

14 selected papers were analyzed considering how many studies were published over the years (MQ1). Figure 3.2 represents this analysis graphically. Although the range of years was not limited in this systematic review and mapping, the first selected paper was from 2005 and all others were published from 2007 onwards. One of the possibilities regarding to a greater number of publications about the use of provenance in the context of SDP appears after 2007 is due to the emergence of the Provenance Challenge[11], started in 2006 (MOREAU *et al*., 2008). However, it should be considered that this event addressed the provenance challenges in the general scope and not

[11] A forum for the provenance community to understand the capabilities of different provenance systems and the expressiveness of their provenance representation (Moreau *et al*., 2008).

specifically in the SDP domain. The results dating from only 2005 also shows the lack of maturity of this research field and the need, as underscored by some authors, of more scientific papers about using provenance in the context of SDP.

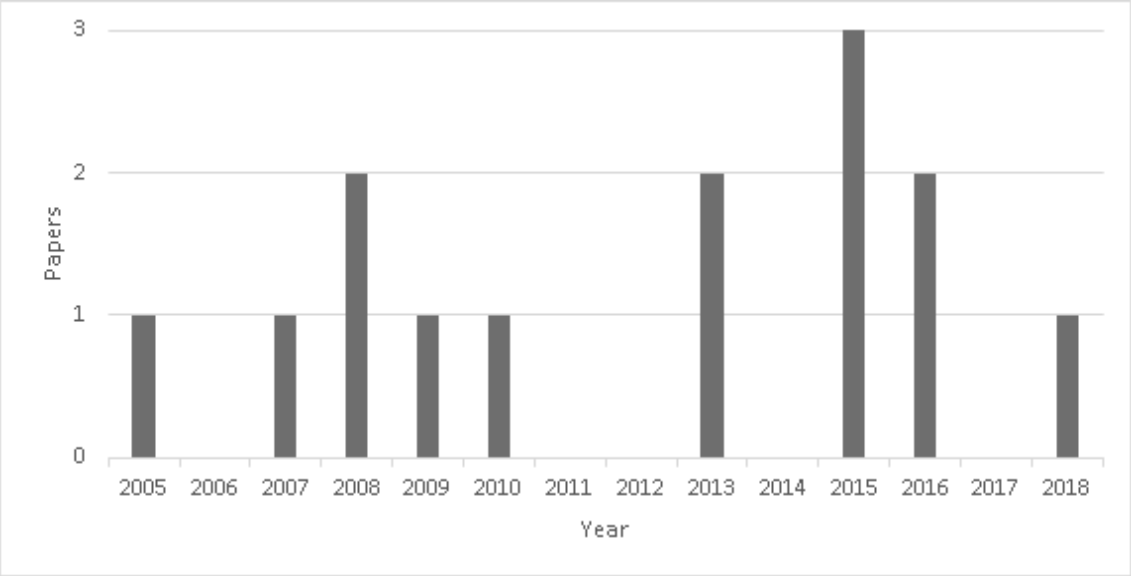


Figure 3.2: Number of papers x year.

The second point analyzed in this mapping was about the most active authors in the review area. Only five researchers appeared more than once in our results and four of them are from the same research group (Gabriella C. B. Costa, Cláudia M. L. Werner, Regina Braga, and José Maria N. David). The authors' name and the total number of related publications are illustrated in Table 3.5.

Table 3.5: Authors' names and number of publications.

Name	Total
Gabriella C. B. Costa, Regina Braga	3
Cláudia M. L. Werner, José Maria N. David, Michael W. Godfrey	2
Abram Hindle, Adrian Cho, Amrit'anshu Thakur, André Luiz de Castro Leal, Andreas Schreiber, Arijit Sengupta, Daniel M. German, Duncan Ruiz, Fernanda Campos, Heinrich Wendel, Humberto L. O. Dalpra, Jaehong Park, José Luis Braga, Julius Davies, Jun Liu, Lianshan Sun, Lin Luo, Maria Luiza Falci, Markus Kunde, Peng Xu, Ping Cheng, Ravi Sandhu, Rayford Vaughn, Rita Cristina Galarraga Berardi, Sérgio Manuel Serra da Cruz, Sudha Ram, Tássio F. M. Sirqueira, Valentine Anantharaj, Victor Stroële, Ya Bin Dang	1

The main publication vehicles type for research production in the review area are exposed in Figure 3.3. Most papers, 7 from a set of 14, were published at conferences (50%), and the other 7 papers were published in journals (21.4%), at workshops (21.4%) and in a seminar (7.1%).

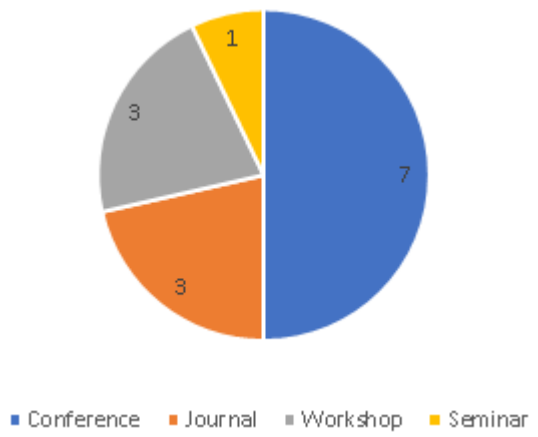


Figure 3.3: Channel type.

In addition to the mapping questions discussed above, a deeper analysis to identify, evaluate and interpret the 14 selected studies to answer the systematic review research questions is needed.

- **Systematic review report**

In order to answer RQ1 (*What are the approaches that apply provenance in SDP domain?*), the approaches name and a brief description about them were identified. A summary of the approaches is presented in Table 3.6. Analyzing and grouping the approaches into specific categories according to their goals is not a trivial task, considering that these goals are not always clearly and directly described in the papers and each approach has its own particularities. Despite this, an attempt of this grouping was made and is presented in the following paragraphs.

Four approaches (***28.6%***) deal with the ***provenance of software artifacts***: Dang *et al*. (2008) consider only the provenance of source code, while provenance of software development artifacts in general is addressed by Xu and Sengupta (2005), Davies *et al*. (2013) and Godfrey (2015).

Other five approaches (***35.7%***) have as main goal making SDP provenance-aware (WENDEL *et al*., 2010) (DALPRA *et al*., 2015) (COSTA, 2016) (COSTA *et al*., 2016b) (FALCI *et al*., 2018), and the last four are focusing on SDP improvement, reusing the experience obtained in previous executions.

The other five remaining approaches (***35.7%***) are very specific, with objectives quite distinct from each other:

- Ram and Liu (2007) propose a model to represent provenance not only in SDP but in various domains;

- Leal *et al*. (2015) map provenance as a catalogue of non-functional requirement;
- Sun *et al*. (2013) propose a framework for access control, based on provenance;
- Berardi and Ruiz (2008) present a framework for evaluating software effort based on provenance data; and
- Thakur *et al*. (2009) address known and unknown vulnerabilities in a system during the test phase of a SDP, using concepts of provenance and pattern matching.

Table 3.6: Identified approaches.

Approach Name	Reference	Approach Description
Ariadne	Dang *et al*., 2008.	A code provenance management tool tracks the provenance of source code and generates provenance reports to facilitate the management of its intellectual property.
FTS (Fully Traceable System)	Xu and Sengupta, 2005.	The approach presents how provenance can be achieved in configuration management by binding an artifact to its traceability and evolution information.
iSPuP (improving Software Process using Provenance)	Costa, 2016.	iSPuP supports measurement definition, execution, monitoring, and analysis of software processes, to improve its performance by using provenance data, ontology, and predefined metrics
OntoComplex	Falci *et al*., 2018	OntoComplex is an architecture that uses ontology, complex networks, and inferences to derive implicit knowledge from provenance data related to software process. The main goal of the architecture, as quoted in the paper, is: "use software process and its execution data analysis, to help managers to make decisions based on acquired knowledge to improve future executions".
PROV-Process	Dalpra *et al*., 2015.	The approach allows the storage and analysis of software process provenance data to identify improvements for future executions of software process instances by using a provenance layer.
Software Bertillonage	Davies *et al*., 2013.	Given a library, file, function, or even snippet of code, this approach determines the entity origin: "was the entity designed to fit into the design of the system where it sits, or has it been borrowed or adapted from another entity elsewhere?".
W7 model	Ram and Liu, 2007.	An ontological model called W7 is presented and represents data provenance as a combination of

		seven interconnected elements including, “what”, “when”, “where”, “how”, “who”, “which”, and “why”.
Leal’s approach	Leal *et al*., 2015.	It introduces the organization of provenance as a catalogue of non-functional requirement (NFR). It considers provenance as a quality factor inherent to the processes (institutions, entities, and activities) that require traceability. According to the authors, this approach enables the construction of chains of operations in software systems to produce pieces of data with higher quality.
Sun’s approach	Sun *et al*., 2013.	The approach is a provenance-aware access control framework with a layered architecture that features an abstract layer, including a Typed Provenance Model (TPM). This model allows the identification, specification, and refinement of provenance-aware access control policies from the beginning of provenance-aware systems development.
Berardi and Ruiz’s approach	Berardi and Ruiz, 2008.	A framework for evaluating software effort data is briefly described. It is divided in four major components: (1) Provenance Component, (2) Inference Machine Component, (3) Quality Database Component, and (4) Provenance and Quality Warehouse Component.
Thakur’s approach	Thakur *et al*., 2009.	A method to address known and unknown vulnerabilities using provenance concepts and pattern matching during the testing phase of a system’s development lifecycle.
Wendel’s approach	Wendel *et al*., 2010.	An approach to make SDP provenance-aware, using a service-oriented architecture to record/store provenance. It uses PRiME (Munroe *et al*., 2006) and OPM. Its main goal is to answer questions related to SDP, such as “Why does the build fail currently?”.
Costa’s approach	Costa *et al*., 2016b.	An approach to support the reuse of experience in previous executions of software processes, using provenance data and ontology. This approach includes the software process enactment, monitoring and analysis improvement using provenance data and ontology and is divided into four distinct layers: (1) Client Layer, (2) Integration Layer, (3) Measure Layer, and (4) Provenance Layer.
Godfrey’s approach	Godfrey, 2015.	The paper analyses the problem of extracting and reasoning about the provenance of software development artifacts. The approach has two distinct phases: (1) a simple metric that is relatively cheap to compute on a large data set, is

	applicable at the level of granularity desired, and has good discriminatory value on candidates, and (2) a more expensive and precise analysis on the result set from the first phase (e.g., an expensive clone detection algorithm might be used that requires deep static analysis of the code, or a manual analysis of the entities is done).

The second research question was: *What are the provenance models for applying provenance in SDP domain?*. All the cited models are shown in Table 3.7. It is important to emphasize that, among the approaches that mention the use of some provenance model, the most used is PROV (MOREAU and GROTH, 2013). Besides that, most of the more recent works uses PROV or are based on it. On the other hand, OPM (MOREAU *et al.*, 2011) is applied by only one approach and is used as the basis for the creation of the Typed Provenance Model (SUN *et al*., 2013). Another important observation is that five approaches (**35.7**%) propose its own provenance model to deal with provenance, so, there is no consensus about the most appropriate model to be used specifically in the SDP domain.

Table 3.7: Identified provenance models.

Provenance Model	Reference
Typed Provenance Model (TPM)	Sun *et al*., 2013.
NFR (Non-Functional Requirement) Catalogue	Leal *et al*., 2015
OPM	Wendel *et al*., 2010.
PROV	Costa *et al*., 2016b. Godfrey, 2015. Costa, 2016. Dalpra *et al*., 2015.
ProvONEExt (an extension of ProvONE)	Falci *et al*., 2018.
SCP Model	Xu and Sengupta, 2005.
W7 model	Ram and Liu, 2007.
Not mentioned	Dang *et al*., 2008. Berardi and Ruiz, 2008. Thakur *et al*., 2009. Davies *et al*., 2013.

In order to answer RQ3 (*What are the benefits that can be achieved by using the approach?*), Table 3.8 was created. The benefits cited in the selected papers are quite varied, indicating that there is no consensus regarding the benefits that can be achieved by using provenance in the SDP domain.

Table 3.8: Approaches benefits.

Reference	Benefits
Dang *et al*., 2008.	Cost reduction of the copyright clearance effort; risk reduction of copyright contamination from external copy-and-paste.
Leal *et al*., 2015.	Enable the construction of chains of operations in software systems to produce pieces of data with higher quality.
Sun *et al*., 2013.	Creation of access control policies from the beginning of provenance-aware systems development; abstraction of complex provenance graphs.
Berardi and Ruiz, 2008.	Allow the company to analyze the present state of the effort data, as well as to identify flawed points and improvement margins.
Thakur *et al*., 2009.	Enable handling of known and unknown exceptions that could be potential threats to the system.
Xu and Sengupta, 2005.	Provide a new method to incorporate versioning, traceability, and provenance in software design.
Wendel *et al*., 2010.	They are not clearly presented; however, it has been inferred that the main benefit of the approach is to record/store provenance data of software development process (using a high level of abstraction), allowing queries.
Davies *et al*., 2013.	The stakeholders can use provenance of software entities information to comply with security standards, licensing, and other software requirements.
Costa *et al*., 2016b.	Provide implicit information to be used for improving process performance, using previously defined metrics.
Godfrey, 2015.	The proposed approach of applying a computationally cheap and conceptually simple matching algorithm to a large data set, then applying a more expensive technique (a manual analysis of the best matches) worked well on the problem of matching library versions identifiers to a large space of possible matches taken from a near-comprehensive master repository.
Ram and Liu, 2007.	The main benefit of the approach is to present a generic model of provenance data and intends to be easily adaptable to represent domain or application specific provenance requirements in active conceptual modeling.
Costa, 2016.	Detection of artifacts that consume more process time, and provide suggestions of how to decrease runtime; Provide support to the software process manager to define process metrics; Provide mechanisms for capturing software process prospective and retrospective provenance; Provide mechanisms of feedback about possible improvements and adjustments to do in the defined process, based on process provenance data and measurements collected during process execution; Provide mechanisms for visualizing process provenance data during the execution, monitoring and analysis phases; Provide mechanisms for deriving implicit information related to process provenance data using ontology and inference machines.
Dalpra *et al*., 2015.	Extract strategic information to the project manager enabling her/him to take decisions that can improve process performance.

Falci *et al*., 2018.	Assist software managers in extracting useful and strategic knowledge from software process data, allowing them to make better strategic decisions about the process.

RQ4 is about the provenance extraction, storing and analysis method (*How was the SDP provenance data extracted, stored, and analyzed?*). Table 3.9 summarizes how the approaches deal with provenance data. Most of the approaches focus on capturing the provenance of software artifacts, although they are manipulated at different levels of granularity (e.g., only source code, classes, any software product etc.). Regarding the storage of provenance data, five approaches (**35.7**%) use relational databases for this purpose. Files with provenance metadata is used by three approaches (DANG *et al*., 2008) (XU and SENGUPTA, 2005) (FALCI *et al*., 2018) and two of them cite the use of a graph database (WENDEL *et al*., 2010) (FALCI *et al*., 2018). Regarding the way of analyzing the provenance data, it can be emphasized that there is no consensus among the analyzed works. The most cited forms were the use of ontologies (in four papers) and a strategy like a technique called "Bertillonage".

Table 3.9: Provenance extraction.

Reference	Artifacts	Provenance storage	Analysis method
Dang *et al*., 2008.	Source code	Metadata file with the same source code name, however, with a different extension (*.orimeta).	They analyze IP metadata to generate IP reports for the specified projects. These reports depict the everyday status of the project's IP pedigree, and project managers and attorneys can review the reports by browser or email. Unsafe items that violated the IP policies can be highlighted for proper actions.
Leal *et al*., 2015.	Not mentioned	Not mentioned	SIG (Softgoal Interdependency Graph)
Sun *et al*., 2013.	Classes, business operations, and actors	Not mentioned	Not mentioned
Berardi and Ruiz,	Not	The framework has a	Not mentioned

2008.	mentioned	Provenance Database	
Thakur *et al*., 2009.	Program statements	It was not mentioned, but, through the text, it appears that a relational database was used to store the provenance data.	Automated clustering based on individual cluster characteristics - put into place some form of clustering technique where 'most similar' candidates appear in the same group. This is performed in a mechanized fashion based on the attributes these candidates possess. Another step is manually aided interpolation to fully define a cluster's elements given its upper and lower bounds.
Xu and Sengupta, 2005.	Software development artifacts in general	XML-based metadata	A component of FTS Architecture called "Inference engine" traces the dependency information in the XML file and suggests the impacted artifacts.
Wendel *et al*., 2010.	Interactions between developers in a distributed tool suite and the resulting artifacts	Graph database (Neo4j)	Graph query language (Gremlin queries)
Davies *et al*., 2013.	Software entities	PostgreSQL database	A technique of software Bertillonage: anchored signature matching.
Costa *et al*., 2016b.	Activities, entities, and agents	MySQL database	An ontology and an inference machine
Godfrey, 2015.	Software entities	Maven2 repository	A strategy that is similar to the metaphor of Bertillonage.
Ram and Liu,	Data objects	Provenance	Not mentioned

2007.	at different granularity levels	annotations	
Costa, 2016.	Activities, entities, and agents	Relational repository	PROV-Process Ontology
Dalpra *et al.*, 2015.	Activities, entities, and agents	PROV-Process relational database	PROV-Process Ontology
Falci *et al.*, 2018.	Process, ProcessExec, Data, and User	The data are stored as ontology individuals (file in OWL format) and using Neo4j3 database management system	Complex network analysis and ontological analysis

The last research question, RQ5, was: *How was the approach evaluated*?. Figure 3.4 shows the obtained result about this question. Most of the approaches present a usage example or a case study as the approach evaluation. It should be emphasized that Davies *et al.*'s (2013) approach presents two types of evaluation: an empirical study and a case study, so the total of approaches shown in Figure 3.4 is 15, instead of 14, which is the number of papers analyzed in this review.

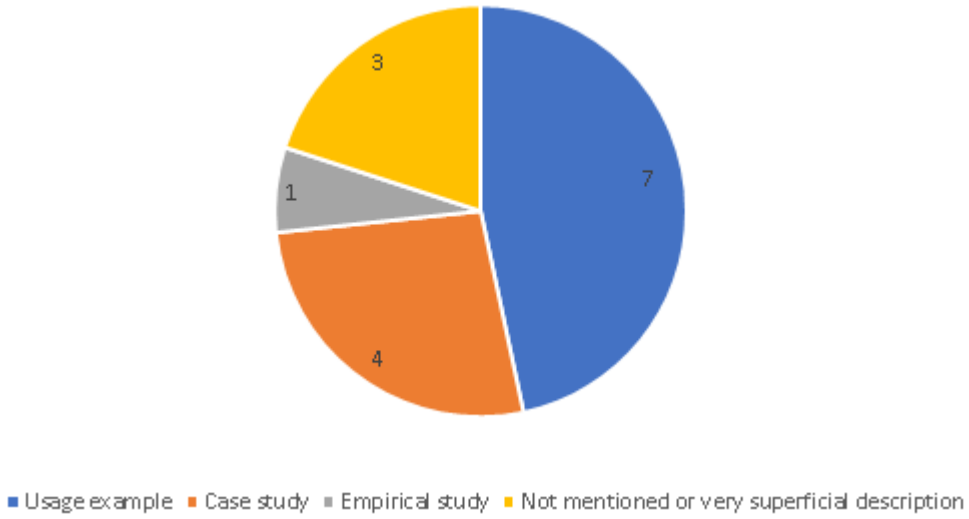


Figure 3.4: Approaches evaluation.

Another analysis carried out in this review was about the quality of the selected papers, using the quality assessment questionnaire (Table 3.3). The results obtained during this analysis are shown in Figure 3.5. With QA1 we assessed if the authors of the study clearly state the aims / objectives of the research. This question could be answered

positively for all the reviewed publications, except for one. With QA2 we asked if the paper clearly explained the proposed approach. For almost publications (64.28%) this could be answered positively. QA3 was checked with "Yes" if the used provenance model was clearly described and its adoption justified. 50% of the papers addressed this issue. QA4 checks if there is any empirical/experimental result regarding the approach, however, most approaches do not provide such type of results. With QA5 we assessed if validity threats were explicitly discussed, however, the scope of validity is scarcely discussed in the selected paper, only 3 presented this type of discussion. Finally, QA6 evaluated whether the selected papers clearly answer to the research questions presented, but, unfortunately, only 35.7% presented clear answers to the paper presented questions.

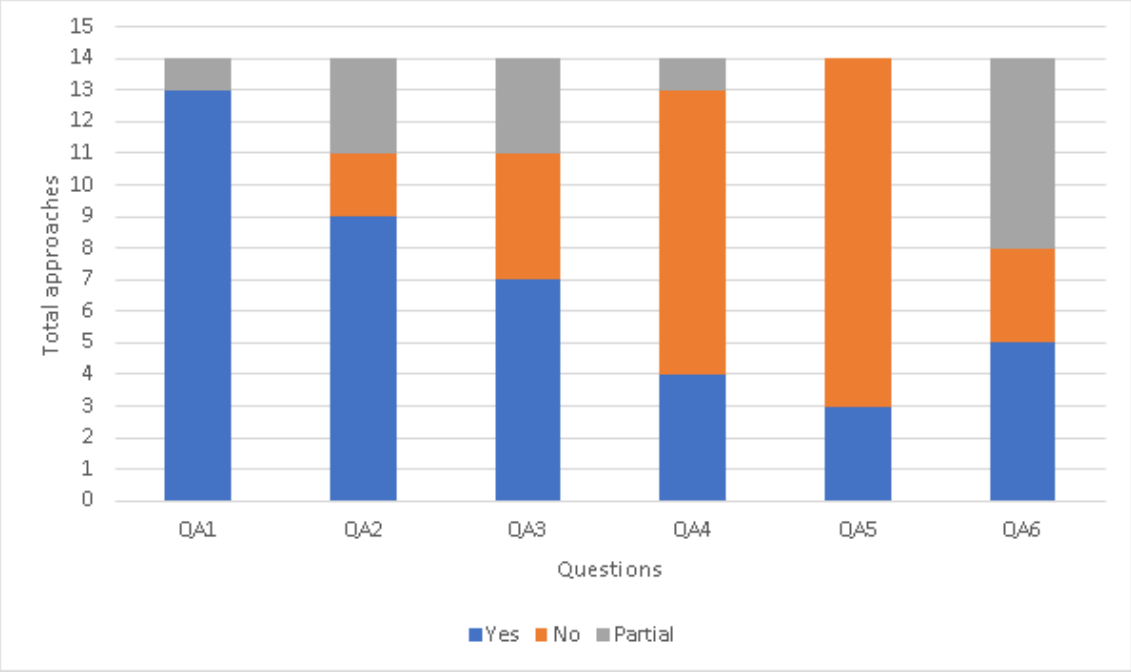


Figure 3.5: Quality assessment results.

3.4 Review Findings and Discussion

The results of the presented literature review confirmed our conjecture that it is still rare in the literature mature proposals addressing the use of provenance in SDP. There is insufficient evidence about the validation of the proposed approaches, which brought up some open questions, and possibilities of future works joining these two themes. In this subsection, the findings of the *quasi*-systematic literature review and mapping are discussed.

- **No proposals before 2005**

Our investigation to answer the mapping question 1 (*MQ1: How many studies were published over the years?*) resulted in 14 approaches and no proposal appeared before 2005. Besides that, no work was found in 2006. These reappear in 2007 to 2010, 2013, 2015, 2016 and 2018. We believe that the development year of these proposals is directly related to the publication of provenance models by the provenance community. In 2006, at the International Provenance and Annotation Workshop (IPAW), the first version of OPM was created during the Provenance Challenge (MOREAU *et al*., 2008) and PROV in 2013, by W3C (GROTH and MOREAU, 2013).

Another observation regarding the distribution of the works over the years is that the works presented in 2018, 2016 and one of 2015 are from our research group (FALCI *et al*., 2018) (COSTA, 2016) (COSTA *et al*., 2016) (DALPRA *et al*., 2015).

- **Research Immaturity**

Considering the publication vehicles and the approaches evaluation methods, we can state that the analyzed research area (provenance in the context of SDP) is still immature. Most papers (11 out of 14) were published in conferences, workshops, or seminars, and only 3 were published in journals. Besides that, considering the approaches evaluation, most of them present only usage examples or simple case studies. Only Davies *et al*. (2013) present an empirical study and a case study as evaluation.

- **Proposals that use provenance in SDP have very different goals**

Although all the analyzed approaches apply provenance in SDP context, they have very different objectives, showing the versatility of issues that can be explored from historical information about SDP data.

The more classical provenance use consists in the analysis of software artifacts production and its respective provenance. Four approaches (DANG *et al*.2008), (XU and SENGUPTA, 2005), (DAVIES *et al*., 2013), and (GODFREY, 2015) deal with provenance of SDP artifacts and the challenges related to the capture, storage, and manipulation of this information.

There are proposals to map provenance as a catalogue of non-functional requirements (LEAL *et al*., 2015), frameworks for access control using provenance information (SUN *et al*., 2013) and software effort evaluation (BERARDI and RUIZ 2008), besides system vulnerabilities detection using concepts of provenance and pattern matching (THAKUR *et al*., 2009).

However, five approaches have a general goal in common: making SDP provenance-aware (WENDEL *et al*., 2010) (DALPRA *et al*., 2015) (COSTA, 2016) (COSTA *et al*., 2016) (FALCI *et al*., 2018), and the last four are focusing on SDP improvement, reusing the experience obtained in previous executions (nevertheless, they are from the same research group).

- **No consensus about the most suitable provenance model**

Although the most widely used model by the analyzed approaches is PROV (according to RQ2), there is no consensus on the most appropriate provenance model to SDP. Most of the approaches (**35.7**%) propose their own provenance models to deal with provenance. Considering these observations and based on the fact that software processes have well-established concepts, a need for standardization of provenance models for this domain is perceived.

- **Different types of benefits**

From the analysis of the selected papers, it can be detected that the benefits when applying provenance in SDP context are quite varied, showing the versatility of benefits that can be obtained when using provenance in the context of SDP. As examples, the following can be cited: (i) cost reduction of the copyright clearance effort (DANG *et al*., 2008), (ii) produce pieces of data with higher quality (LEAL *et al*., 2015), (iii) creation of access control policies using provenance (SUN *et al*., 2013.), (iv) analyze software effort data, as well as to identify flawed points and improvement margins (BERARDI and RUIZ 2008), (v) handling of known and unknown exceptions (THAKUR *et al*., 2009), (vi) incorporate versioning, traceability, and provenance in software design (XU AND SENGUPTA, 2005), (vii) record/store provenance data of SDP allowing its querying (WENDEL *et al*., 2010), (viii) improving SDP (COSTA, 2016) (COSTA *et al*., 2016), and (ix) assist process managers in decision-making (DALPRA *et al*., 2015) (FALCI *et al*., 2018). The benefits cited by all approaches are presented in detail in Table 2.9.

- **Empirical studies scarcity, low rigor and relevance values**

Based on the answer about RQ5 (*M*ost of *the approaches present only usage examples or simple case studies, and only one has an empirical study*, presented in Section 6.2) and in the quality assessment questionnaire and its results (Figure 2.11), this review shows the scarcity of empirical studies and low rigor and relevance values when considering the application of provenance in software processes. We believe that a greater rigor in evaluating the approaches benefits in real environments is still

necessary, in order to show real-world evidence of these benefits for software development companies.

3.5 Threats to Validity

It is also important to consider the threats to validity and limitations of this quasi-systematic literature review and mapping. The results presented in this review may have been influenced by certain uncontrollable limitations. Regarding internal threats to validity, it should be mentioned that in the paper evaluation filters only two researchers analyzed the results and, when there was disagreement among the opinions, the papers were included for analysis in the next step. Another point that should be considered is that data extraction was performed by only one researcher (the other just reviewed the extraction forms after they were filled out), which may entail some risk of bias. Additionally, the search string may not contain all the relevant keywords causing loss of some valuable studies. However, the search string was evaluated using papers to control the obtained results. Furthermore, all the works that were found in a previous informal literature review were returned and analyzed in this quasi-systematic literature review and mapping, generating evidence about the search string correctness. Some electronic databases such as Springer Link and ACM Digital Library were not considered in this paper, taking into account the criteria exposed in Subsection 4.4, so it is possible that relevant studies were not indexed by our selection. However, we believe that the selected electronic databases were enough to obtain a picture of the use of provenance in the context of SDP.

3.6 Final Remarks

The presented quasi-systematic literature review and mapping aimed to identify and investigate in detail how provenance has been applied in SDP. During the review and mapping, we started with 125 papers identified by the selected electronic databases, which were filtered, and resulted in 14 selected papers.

Provenance data began to be applied in the domain of SDP after 2005 and there are few researchers or groups of researchers working in these two areas (provenance AND software processes). Only two authors appeared more than twice in the paper selection and with 3 publications. In addition, most of the selected papers (50%) were published at conferences.

Analyzing the content of the selected papers, 28.6% of them do not consider or do not deal with the provenance of SDP, e.g., they deal only with the provenance of software "artifacts". On the other hand, there are approaches (35.7%) focused on making SDP provenance-aware. There is no consensus regarding the benefits that can be achieved by using provenance in the SDP domain and about the most appropriate provenance model to be used specifically in SDP domain. The most used is PROV (28.6%) and other 35.7% propose its own provenance model. Few studies focused on developing and evaluating a concrete proposal about the topic of this review. Only one work answered 'Yes' for all the checklist quality questions. One possible reason for this was that most part of the analyzed papers does not present a rigorous or detailed description and evaluation about the proposed approach. Such findings generate evidence that, although many studies indicate the need and the possibility of obtaining several advantages through the application of provenance techniques in the field of software development processes, it is still rare to find proposals that have been implemented and evaluated through experimental studies.

Finally, it is important to cite that this thesis fits in Software Analytics area, considering it can help managers in answering important questions about their process. Software Analytics can be defined as "*analytics on software data for managers and software engineers with the aim of empowering software development individuals and teams to gain and share insight from their data to make better decisions*" (BUSE and ZIMMERMANN, 2012). However, our approach differs from other software analytics applications and techniques (MENZIES and ZIMMERMANN, 2013) considering the following points: (i) they do not involve a provenance model that deals with SDP specificities, bringing more accuracy to process data; (ii) they do not use semantic models, like ontologies and intelligent mechanisms, such as inference machines to derive new knowledge from these data as in iSPuP approach. Therefore, in the next chapter, we present the PROV-SwProcess, the core model of iSPuP approach.

CHAPTER 4 – PROV-SwProcess PROVENANCE MODEL

This chapter presents PROV-SwProcess model, the provenance model developed to accommodate SPD provenance specificities, including its main elements, relations, inference rules and competency questions. PROV-SwProcess is the core of iSPuP approach.

4.1 Introduction

Based on the literature review presented in Chapter 3, there is no consensus regarding the most appropriate provenance model to be used specifically in SPD domain. The model most used in the provenance area is PROV. However, the direct application of this model to SPD domain lacks in capturing some SPD specificities such as Resources and Procedures used or adopted by the activities, different types of SDP artifacts (e.g., software product, software items and models), as well as new possible relationships between them. To overcome this gap, an extension model for SDP provenance representation is proposed, named PROV-SwProcess model. This model was defined as an extension of PROV model, aiming to capture and store relevant information about SDP provenance data. Besides that, considering the existence of different applications that can be used during SP execution (e.g., version control system, issue trackers, and documentation management systems) and different models and formats adopted by these applications, PROV-SwProcess model was defined, in order to be used as a standard model to them. In addition to capturing / storing provenance data, this model provides a structure that allows a better analysis from software process provenance data later.

Before the specification of PROV-SwProcess, next subsection presents the relation of the proposed model with other provenance models (or provenance model extensions) already existing in the literature. Afterwards, all the aspects covered by PROV-SwProcess are listed and, finally, the complete description of the model is presented.

4.2 Relation with other standards / models

PROV-SwProcess was developed as an extension of PROV model to capture SDP provenance data. This extension has been developed considering that PROV is more general but does not provide all concepts related to SDP.

PROV-SwProcess uses as basis for its definition the package of Software Process Execution of the Software Process Ontology (SPO) (FALBO and BERTOLLO, 2009). This ontology establishes a common conceptualization about the software process domain and includes processes, activities, resources, people, artifacts, and procedures.

Another extension of PROV model specification is D-PROV (MISSIER *et al*., 2013). It has the aim of representing process structure, i.e., to enable the storage and query using prospective provenance. Missier *et al*. (2013) show an example of using D-PROV in the context of scientific workflows. D-PROV was a previous version of ProvONE (CUEVAS-VICENTTÍN *et al*., 2016). ProvONE is a model for scientific workflow provenance that extends PROV with its specific structure elements. It was developed in the context of DataONE Project (DATA ONE, 2018), a large scale and federated data infrastructure for the earth sciences community. Although this model is useful in the scientific workflow domain, it is not adequate for capturing and analyzing provenance in the software development process domain. For example, in ProvONE, the workflow execution corresponds to the execution of computational tasks only by software agents but, in the software process context, we need to express different types of agents (called *stakeholders* in SDP), such as, person, teams and organizations, besides the use of software agents as *resources*. ProvONE provides two ways to represent the data manipulated during the workflow execution: using Data class or Collection class, however, in SDP context, we have different data types that could be *used*, *generated*, *modified*, or *adopted* during some process instance, e.g., artifacts (Software Products, Software items, Document, Models, Information items), procedures (methods, templates, techniques), and resources (hardware and software). Besides that, PROV-SwProcess offers several causal relations that are not expressed in ProvOne, e.g., *wasBasedOn*, *wasAppliedTo*, *hadRole*, *created*, *modified*, etc., created and evaluated by experts in provenance and software process (Chapter 6) to accommodate SDP specificities.

Versioned-PROV (PIMENTEL *et al*., 2018) proposed a PROV extension to accommodate mutable data structures, using reference derivations and checkpoints, allowing to represent multiple versions of a data object. This approach was proposed considering that PROV assumes immutable entities, and all changes to an entity are represented by the creation of a new entity. Then, it could cause an overhead on the provenance storage, when dealing with fine-grained provenance.

A preliminary proposal of PROV-SwProcess (called PROV-Process) was published in Dalpra (2016). It is a master thesis developed in the context of this doctoral thesis. It consists in an initial approach to apply the PROV model in SDP domain. PROV-SwProcess aims to incorporate the basic ideas of PROV-Process, as well as additional contributions (new constructors - *Software_Process*, *Procedure*, *Resouce*, and its respective subtypes; *Entities* are renamed to *Artifacts* and five specific artifacts subtypes were included; new relations between these constructs were specified, and eight groups of inference rules were carefully defined and implemented), to derive an adequate model that can be used in the SDP.

PROV-SwProcess model presented in this thesis is in its third version[12] (the first version[13] was evaluated by two experts in software processes and provenance; after that, a second version[14] was generated using the expert's corrections and suggestions. Finally, a third expert evaluated this second version and we created the current version of PROV-SwProcess model). More details about PROV-SwProcess evaluation with experts are presented in Chapter 6.

4.3 Aspects covered by PROV-SwProcess

PROV-SwProcess aims to provide the fundamental information required to understand and analyze provenance data from SDP. Considering this, it covers prospective and retrospective provenance (FREIRE *et al*., 2008):

- **Prospective provenance*:*** captures the steps, i.e., the procedure necessary to produce the software. It corresponds to the SDP specification, detailing all the activities (and/or sub activities) that must be carried out to generate the software. In addition, it specifies the responsible(s) for the process, the specific roles to perform an activity, the procedures, and resources to be

[12] Available at: http://www.gabriellacastro.com.br/provswprocess/v3.html
[13] Available at: http://www.gabriellacastro.com.br/provswprocess/
[14] Available at: http://www.gabriellacastro.com.br/provswprocess/v2.html

adopted, and artifacts to be generated, used, or modified during the process execution. In order to enable a degree of abstraction, this specification does not need to be executable.

- **Retrospective provenance*:*** comprises the activities that were executed in the specification / implementation of the software, considering the adopted procedure, the artifacts generated, altered, and / or used, the stakeholders involved, and the resources used during the process execution. This information can be recorded at varying degrees of detail and granularity, depending on how the recording is treated on the software process development system.

Besides that, PROV-SwProcess includes the essential aspects of SDP: activities, stakeholder, resource, procedure, and artifact, as proposed in SPO (FALBO and BERTOLLO, 2009):

- **Activity*:*** deals with the process activities used to create and/or maintain software and how they compose the software development process.
- **Stakeholder*:*** refers to organizations, persons, projects, or teams acting or interested in the software process activities.
- **Resource*:*** involves hardware equipment and software products used by the software process activities.
- **Procedure*:*** relates to methods, techniques and document templates adopted by the software process activities.
- **Artifact*:*** represents different types of objects produced, changed, and used in process activities.

4.4 PROV-SwProcess Model Specification

PROV-SwProcess is divided into (i) associations (or relations), (ii) classes, and (iii) specific inference rules, in order to allow relevant SDP data capturing. Figures 4.1 to 4.4 show four diagrams to represent PROV-SwProcess conceptual model. The following points should be considered when analyzing these diagrams:

- Constructs and associations presented between "<<>>" were derived from PROV. For example: the <<Activity>> class corresponds to the Activity PROV type. Newly PROV-SwProcess associations / relations and classes appeared without "<<>>";

- Elements in yellow ellipses are specializations of the Entity PROV type and elements in orange pentagons are specializations of the Agent PROV type;
- Associations with black solid lines are used to capture Retrospective Provenance, associations with blue solid lines are used to capture Prospective Provenance, and associations with red dashed lines can be inferred by PROV-SwProcess approach and their respective provenance rules, that is, they do not necessarily need to be captured or informed in the SDP provenance data.
- All PROV-SwProcess relations have a related inverse relation (for example: the inverse relation of <<Used>> is the relation <<WasUsedBy>>), however, these were not explicit in the figures aiming to facilitate the understanding of the model;

In order to use the model, SDP data must be loaded into the model. This SDP data is called an SDP instance in this thesis. Figure 4.1 presents PROV-SwProcess model constructs, considering its Retrospective Provenance part. Besides that, when there is more than one instance of performed process to be analyzed, the relation *WasComposedBy* can also be inferred, as show in Figure 4.2, allowing to obtain all stakeholders, resources, artifacts, and procedures involved in a specific performed process.

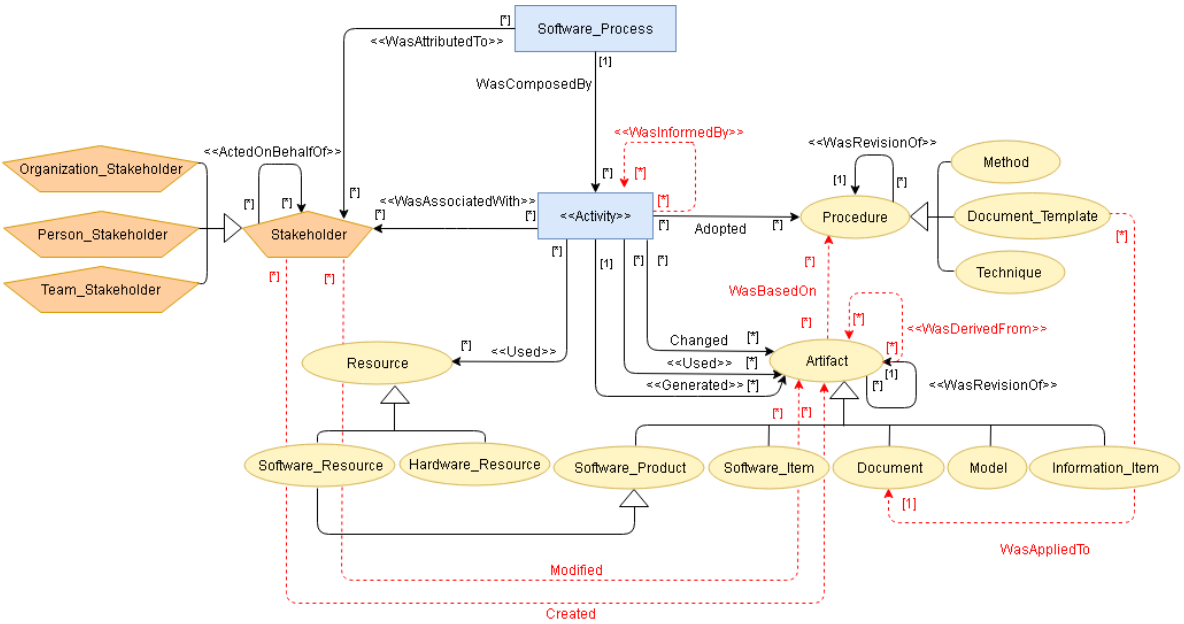


Figure 4.1: PROV-SwProcess - Retrospective Provenance (Part 1) - Conceptual Model.

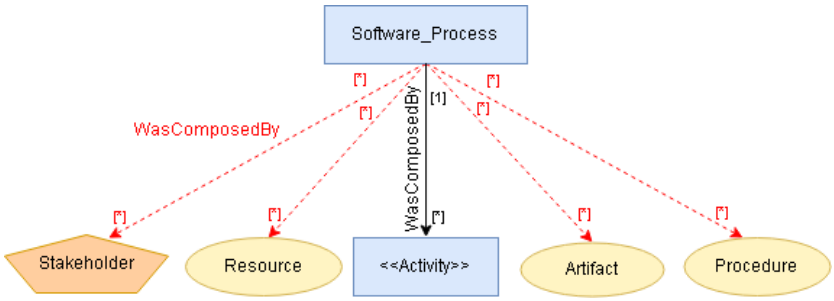


Figure 4.2: PROV-SwProcess - Retrospective Provenance (Part 2) - Conceptual Model

PROV-SwProcess model deals with three levels of any software process instance: defined process, instantiated process and executed process. The retrospective part of the model aims to capture data from executed process. Then, the other two levels are treated by PROV-SwProcess Prospective Provenance. Based on this, Prospective is divided into two parts in PROV-SwProcess: (i) Standard Process Level, and (ii) Intended Process Level. Figure 4.3 presents PROV-SwProcess model constructs, considering Standard Process Level and Figure 4.4 presents the constructs of Intended Process Level.

All classes and relationships presented in Figures 4.1 to 4.4 are presented in detail in the complete specification of the PROV-SwProcess model[15].

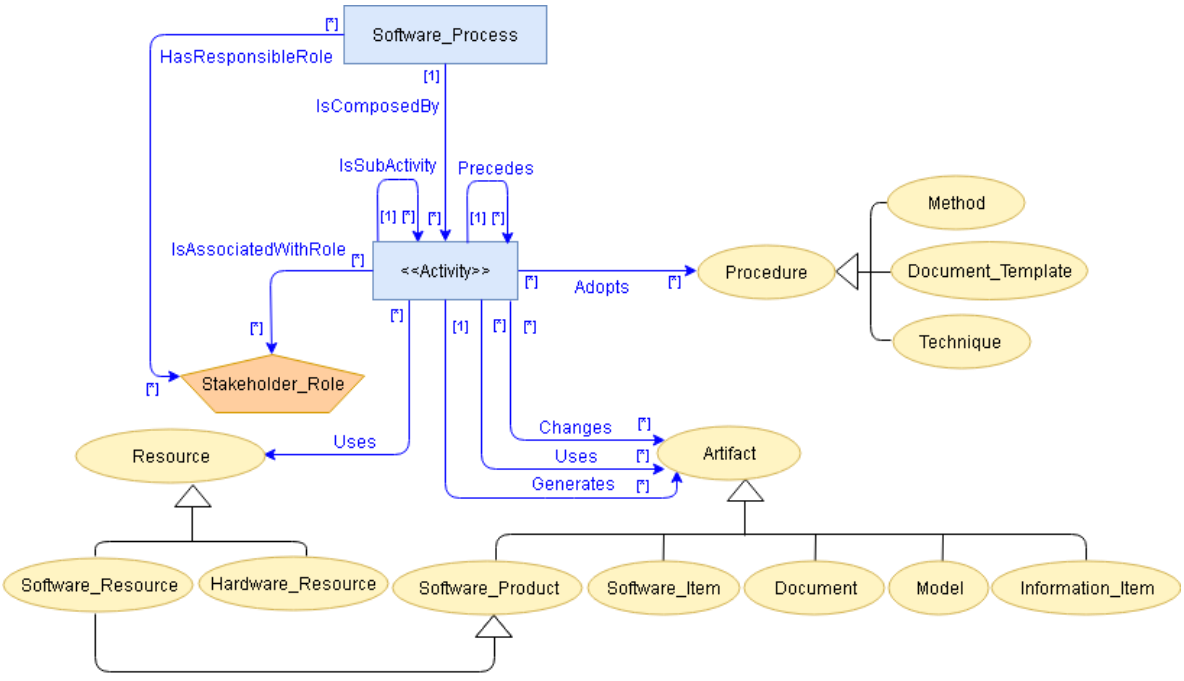


Figure 4.3: PROV-SwProcess - Prospective Provenance of Standard Process Level

[15] Available at: http://www.gabriellacastro.com.br/provswprocess/v3.html

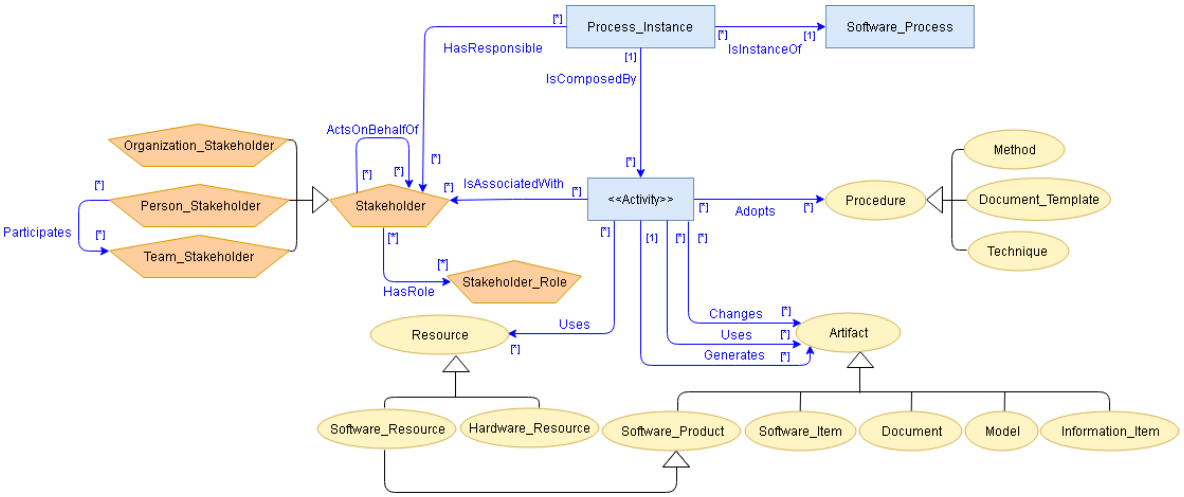


Figure 4.4: PROV-SwProcess - Prospective Provenance of Intended Process Level.

PROV-SwProcess model has also an operational ontology[16] that extends PROV-O ontology (LEBO *et al.*, 2013), and is specified using OWL2 (W3C, 2012). This ontology has specific inference rules that may be used on SDP provenance data. An inference rule can be applied to PROV-SwProcess instances to add new PROV-SwProcess statements, bringing implicit information. The following subsection presents these rules in detail.

4.4.1 PROV-SwProcess Inference Rules

PROV model presents in its documentation some constraints (DE NIES, 2013). They define when a PROV instance is valid, ensuring that this instance represents a consistent history of objects and their interactions are safe to use for logical reasoning or for other types of analysis. Part of this document describes the inferences that may be used on provenance data.

Considering an inference as a rule that can be applied to PROV-SwProcess instances to add new PROV-SwProcess statement, PROV-SwProcess model also specifies its inference rules. In addition to specifying them, they are also implemented in the PROV-SwProcess Ontology.

Eight groups of inference rules have been defined and specified using the Semantic Web Rule Language (SWRL) (HORROCKS *et al.*, 2004), specifically to the SDP domain:

1. Created
2. Modified

[16] PROV-SwProcess Ontology: http://gabriellacastro.com.br/provswprocess/provswprocess.owl.

3. WasBasedOn
4. WasAppliedTo
5. WasDerivedFrom
6. WasInformedBy
7. WasComposedBy
8. HadRole

After the inference definition using SWRL, an example of its operation is presented.

1. Created

```
prov:wasAssociatedWith(?ac, ?sta) ^ prov:generated(?ac, ?art) -> provswprocess:created(?sta, ?art)
```

This inference states that if an activity ***ac*** was associated with a stakeholder ***sta*** and this activity ***ac*** generated an artifact ***art***, the relation ***created*** between the stakeholder ***sta*** and the artifact ***art*** can be inferred.

Figure 4.5 shows an example to explain PROV-SwProcess model possible inferences (the inferred associations appear in red). Even if there is no explicit and direct relation in the provenance data between *Mary* and *Payment_Test_Cases*, we can infer, using the rule presented by Inference 1, that *Mary* ***created*** *Payment_Test_Cases*.

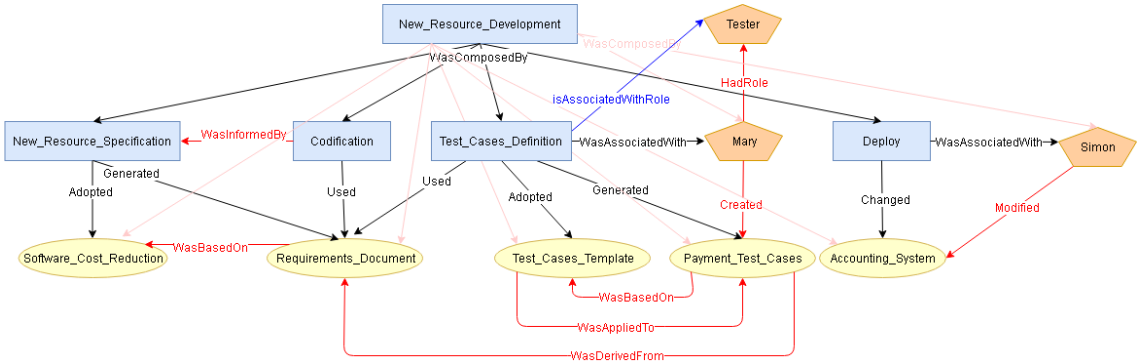


Figure 4.5: PROV-SwProcess Inferences Example.

2. Modified

```
prov:wasAssociatedWith(?ac, ?sta) ^ provswprocess:changed(?ac, ?art) -> provswprocess:modified(?sta, ?art)
```

This inference states that if an activity ***ac*** was associated with a stakeholder ***sta*** and during this activity ***ac*** an artifact ***art*** was changed, the relation ***modified*** between the stakeholder ***sta*** and the artifact ***art*** can be inferred.

Figure 4.5 shows that even if there is no explicit and direct relation in the provenance data between *Simon* and *Accounting_System*, we can infer, using the rule presented by Inference 2, that *Simon* ***modified*** the *Accounting_System*.

3. WasBasedOn

```
provswprocess:adopted(?ac, ?pro) ^ prov:generated(?ac, ?art) -> provswprocess:wasBasedOn(?art, ?pro)
provswprocess:adopted(?ac, ?pro) ^ provswprocess:changed(?ac, ?art) -> provswprocess:wasBasedOn(?art, ?pro)
```

This inference states that if an activity ***ac*** adopted a procedure ***pro*** and this same activity ***ac*** generated or changed an artifact ***art***, the relation ***wasBasedOn*** can be inferred between the artifact ***art*** and the procedure ***pro***.

Figure 4.5 shows that even if there is no explicit and direct relation in the provenance data between *Payment_Test_Cases* and *Test_Cases_Template*, we can infer, using the rules presented by Inference 3, that *Payment_Test_Cases* ***wasBasedOn*** *Test_Cases_Template*. Another inference of this same type can be seen between *Requirements_Document* and *Software_Cost_Reduction*.

4. WasAppliedTo

```
provswprocess:adopted(?ac, ?dt) ^ prov:generated(?ac, ?d) -> provswprocess:wasAppliedTo(?dt, ?d)
provswprocess:adopted(?ac, ?dt) ^ provswprocess:changed(?ac, ?d) -> provswprocess:wasAppliedTo(?dt, ?d)
```

These inferences state that if an activity ***ac*** adopted a document template ***dt*** (a specific type of procedure) and this same activity ***ac*** generated or changed a document ***d*** (a specific type of artifact), the relation ***wasAppliedTo*** can be inferred between the document template ***dt*** and document ***d***.

Figure 4.5 shows that even if there is no explicit and direct relation in the provenance data between *Test_Cases_Template* and *Payment_Test_Cases*, we can infer, using the rules presented by Inference 4, that *Test_Cases_Template* ***wasAppliedTo*** *Payment_Test_Cases*.

5. WasDerivedFrom

```
prov:used(?ac, ?art1) ^ prov:generated(?ac, ?art2) -> prov:wasDerivedFrom(?art2, ?art1)
```

This inference states the derivation between two artifacts if an activity ***ac*** has used an artifact ***art1*** and this same activity generates a new artifact ***art2***.

When this inference was implemented in the SDP domain, it allowed inferring when an artifact was derived from another, although this relation was not explicit in the provenance data. As can be seen in Figure 4.5, we can infer that *Payment_Test_Cases* ***wasDerivedFrom*** *Requirements_Document*.

6. WasInformedBy

```
prov:used(?ac2, ?art) ^ prov:generated(?ac1, ?art) -> prov:wasInformedBy(?ac2, ?ac1)
provswprocess:changed(?ac2, ?art) ^ prov:generated(?ac1, ?art) -> prov:wasInformedBy(?ac2, ?ac1)
```

These inferences state that if an activity ***ac2*** used or changed an artifact ***art*** that was generated by an activity ***ac1***, the relation ***wasInformedBy*** can be inferred between ***ac2*** and ***ac1***, stating a dependency between these activities. Figure 4.5 shows that even if there is no explicit and direct relation in the provenance data between the activities *Codification* and *New_Resource_Specification*, we can infer, using the rules presented by Inference 6, that *Codification* ***wasInformedBy*** *New_Resource_Specification*.

7. WasComposedBy

```
provswprocess:wasComposedBy(?sp, ?ac) ^ prov:wasAssociatedWith(?ac, ?sta) -> provswprocess:wasComposedBy(?sp, ?sta)
provswprocess:wasComposedBy(?sp, ?ac) ^ provswprocess:changed(?ac, ?art) -> provswprocess:wasComposedBy(?sp, ?art)
provswprocess:wasComposedBy(?sp, ?ac) ^ prov:generated(?ac, ?art) -> provswprocess:wasComposedBy(?sp, ?art)
provswprocess:wasComposedBy(?sp, ?ac) ^ prov:used(?ac, ?art) -> provswprocess:wasComposedBy(?sp, ?art)
provswprocess:wasComposedBy(?sp, ?ac) ^ prov:used(?ac, ?res) -> provswprocess:wasComposedBy(?sp, ?res)
provswprocess:wasComposedBy(?sp, ?ac) ^ provswprocess:adopted(?ac, ?pro) -> provswprocess:wasComposedBy(?sp, ?pro)
```

These inferences are only useful when more than one process instance is being analyzed. It is very important because it brings all the *Stakeholders*, *Resources*, *Artifacts*, and *Procedures* of a given SDP instance, when dealing with multiple SDP instances, i.e., we can analyze which process elements participated in only one process instance or in various.

Figure 4.5 shows that even if there is no explicit and direct relation in the provenance data between the SDP *New_Resource_Development* and all the *Stakeholders*, *Resources*, *Artifacts*, and *Procedures*, using this inference is possible to obtain a direct association between them.

8. HadRole

provswprocess:isAssociatedWithRole (?ac, ?r) ^ prov:wasAssociatedWith(?ac, ?sta) -> provswprocess:hadRole(?sta, ?r)

This inference encompasses both PROV-SwProcess retrospective provenance (using the relations *wasAssociatedWith* and *hadRole*) and prospective provenance (*isAssociatedWithRole* relation). It states that if an activity ***ac*** is previously associated with some role ***r***, and a stakeholder ***sta*** has been involved in this activity ***ac*** during its execution, the relation ***hadRole*** between the stakeholder ***sta*** and the role ***r*** can be inferred.Figure 4.5 shows that even if there is no explicit and direct relation stating that Mary acted as a *Tester*, using this inference it is possible to obtain a direct association (*HadRole*) between the role *Tester* and the stakeholder *Mary*.

4.5 PROV-SwProcess Competency Questions

A competency question (CQ) (USCHOLD and GRUNINGER, 1996) is a natural language sentence that expresses a pattern for a type of questions that people / computational applications expect an ontology to answer. Then, the answerability of CQs can be considered as functional requirements of an ontology. Therefore, any approach that uses an ontology as a knowledge base, must use CQ to verify if the ontological goals are achieved.

To demonstrate the potential of PROV-SwProcess Model and its respective operational ontology, a series of CQs were developed. These questions are designed to prove PROV-SwProcess ability in answering questions about SDP, based on provenance and execution data[17]. Besides that, the relevance of each CQ is evaluated by three process managers. We believe that the answers to these questions can assist in the SDP analysis and decision-making (an evaluation with real data and feedback from process managers show initial evidence of this in Chapter 7).

[17] We assume that other questions can be derived from these questions and from the model. This is just an initial set whose relevance was initially evaluated in an interview with process managers (presented in Chapter 7).

A discussion about how PROV-SwProcess model and its tool support can help in answering these questions and some insights of how to use them in SDP analysis and decision-making are detailed after each CQ.

Competency questions are divided according to three main goals:

- **Goal 1**: Process Structure Identification and possibilities for process redesign;
- **Goal 2**: Understanding stakeholder's involvement in process execution; and
- **Goal 3**: Tracking derivations and revisions among artifacts or procedures.

All CQs are detailed in the following.

Goal 1: Process structure identification during execution and possibilities for process redesign

– **CQ1** *What are the process activities, artifacts, resources, procedures, stakeholders, and the relations among them during the process execution?*

- **How can our approach help in answering this question?** Using a list or a graph with the executed activities, artifacts, resources, procedures, and stakeholders with its respective relations.
- **Analysis**: It is possible to identify all the process elements that participated in process executions and the relation among them.
- **Decision-Making Possibility**: After identifying the process elements and the relations between them, it is possible to find gaps (elements without association or inadequate relation established) in the analyzed data and try to correct it in future process executions or change the process model specification.

An example of how PROV-SwProcess helps in answering CQ1 is shown in Figure 4.6. A provenance graph is used, and, in the left corner of this figure, it could be verified that there is a stakeholder (represented by an orange pentagon), without any specific name (NULL). Figure 4.7 shows a tooltip when hovering the mouse on this specific stakeholder. Considering this fact, it could be identified that during the execution of the analyzed process, some problem occurred, and the associated stakeholder with some activities (*Issue_Resolution_3281* and *Issue_Resolution_3583*, for example) were not established. This analysis shows that there is some flaw possibility in the execution of the analyzed process, considering it should not be allowed to execute an activity without associating it with a specific stakeholder (this information was provided by the process manager).

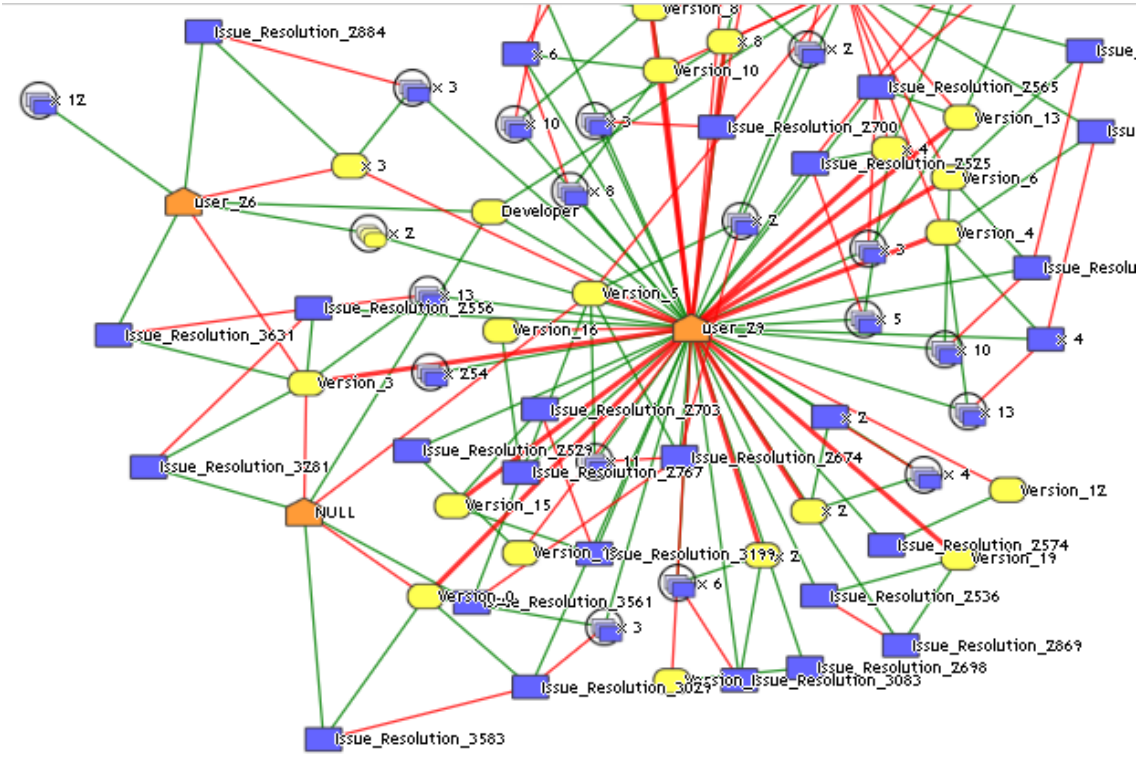


Figure 4.6: Example of provenance graph to support CQ1.

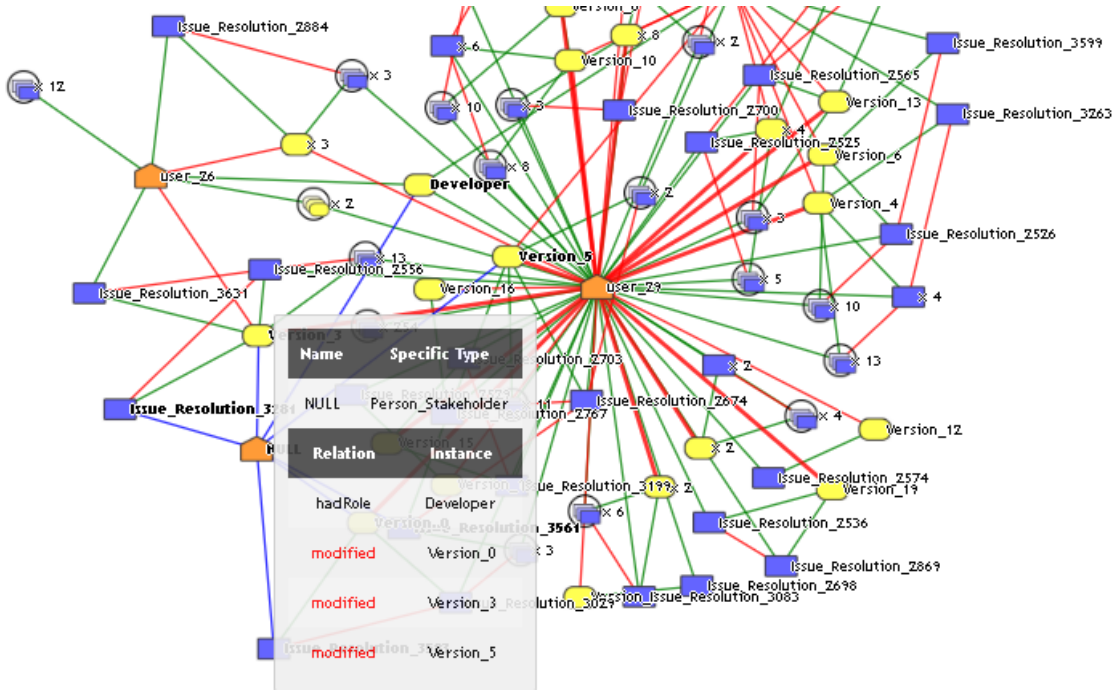


Figure 4.7: Tooltip when hovering the mouse on *NULL* Stakeholder.

All other competence questions are presented in the following. However, a complete example of their utilization is detailed in Section 5.4, after presenting iSPuP approach details to support SDP analysis and decision-making.

– **CQ2:** *Which procedures are used by the process during its execution?*

- **How can our approach help in answering this question?** Using the number of procedures used to develop the process artifacts and a list or graph with them.
- **Analysis**: It is possible to check which procedures influenced an artifact development; Verify the procedures most useful in the analyzed instance(s), when a procedure is used by artifacts in a number greater than the average; Check procedures useless, i.e., although existing, these procedures were never used during the execution of the processes carried out by the organization.
- **Decision-Making Possibility**: When verifying that procedures influenced an artifact development, the process manager can evaluate if this fact was really planned/expected (in the process modeling phase) or not; if this information is not specified in the process model, the process manager may include it; Being aware that a procedure is widely used by the process executions, the manager can better plan any changes in this procedure (when it will necessary), since this can have a great impact on future executions; If a procedure has not been used during process execution, this in-formation may be valid for the process manager to evaluate whether this procedure needs to be changed/reshaped to be used as planned or if it should be removed from the process model. Another point of analysis would be the impact of not having a standard for the development of some artifacts – it could impact the quality level of generated artifacts, as well as cause errors by the difficulty of understanding some information in these artifacts, etc.

– **CQ3:** *Which activities had a high complexity (considering the number of associated stakeholders, artifacts, procedures and / or resources)?*

- **How can our approach help in answering this question?** Using the number of Artifacts, Stakeholders, Procedure and Resource associated to a specific activity or a graph showing these relations.
- **Analysis**: It is possible to check when activities are associated with many stakeholders, artifacts, procedures and / or resources, when compared to the other activities of the process, indicating that this activity could be more complex than others.
- **Decision-Making Possibility**: With the information provided by the analysis presented above, the process manager can evaluate if this fact was really planned/expected (in process modeling phase) or not; if this information is not

specified in the process model, the process manager may change the process model to better represent the process that was in fact executed; A possible evaluation of the activities detected as more complex can be performed, aiming to divide it into less complex sub activities.

– **CQ4:** *Which activities had a high dependency (on other activities)?*

- **How can our approach help in answering this question?** Using the number of dependent activities of each executed activity and a list of them or some graph representation showing these dependencies.
- **Analysis**: It is possible to analyze the dependency between two activities, i.e., when the exchange of some artifact by two activities, one activity using some entity generated or changed by the other occurred. It is also possible to discover which activity occurred before or after another during execution time and to identify possible bottlenecks based on activities dependency.
- **Decision-Making Possibility**: From the previous analyzes, the process manager can confront the activities (and its flow) specified in the process model and how they occurred during execution. If there is any discrepancy, he/she can make changes in the process model, according to what he/she verified that, in fact, was executed. Another decision is trying to make chances in the process model in order to avoid bottlenecks, if it were identified in the previous analysis.

Goal 2: Understanding stakeholder's involvement in process execution

– **CQ5:** *What is the activities distribution among stakeholders?*

- **How can our approach help in answering this question?** Using the number of activities each stakeholder is involved and a list of them or some graph representation showing activities x stakeholders.
- **Analysis**: It is possible to discover, from a stakeholder, all the activities (and the total of these activities) in which he/she participated, allowing to understand the activities distribution among stakeholders in the process execution.
- **Decision-Making Possibility**: When verifying that a stakeholder is participating in more activities than others, the process manager can evaluate if this fact was really planned/expected (considering, for example, that a stakeholder was associated to a high number of activities because him/her always is attributed to activities with a

lower level of complexity) or if it has been occurring due to an inadequate activity distribution during the process instantiation.

- **CQ6:** *Which artifacts are known by a stakeholder, considering that in some process execution he/she created or modified such artifact?*
  - **How can our approach help in answering this question?** Using the number of artifacts each stakeholder is involved in its creation or modification and a list with them.
  - **Analysis**: It is possible to discover all the artifacts that were created and / or modified by a stakeholder, allowing to understand about what artifacts this stakeholder has some knowledge, considering he/she manipulated this artifact in some process execution. Considering the artifact viewpoint, it is possible to discover all the stakeholders that have some knowledge about it, considering it was created or modified by them.
  - **Decision-Making Possibility**: in a future execution of the analyzed process, if a certain task is associated with a specific artifact, the process manager (or the responsible for the process instantiation) can allocate to this task a stakeholder with greater or less knowledge about the artifact to be manipulated during this task execution, according to the project objectives / goals.
- **CQ7:** *Which stakeholders are out of the average of created and/or modified artifacts?*
  - **How can our approach help answering this question?** Based on the artifacts manipulation average and on the number of artifacts each stakeholder is involved in its creation or modification.
  - **Analysis**: It is possible to discover, from a stakeholder, the total of and which artifacts were created or modified by him/her, allowing to understand the performance of this stakeholder considering the manipulation of process artifacts (e.g., if he/she usually creates new artifacts or if he/she only modified them).
  - **Decision-Making Possibility**: When verifying that a stakeholder is creating much artifacts than others, the process manager can evaluate if they really need to be created or if there is a stakeholder's lack of knowledge about the existing and available artifacts to be changed/adapted; the process manager can better specify the responsible for the artifacts manipulation in a future execution of the analyzed

process in order to obtain a better balance in relation to the stakeholder performance, considering the number of artifacts handled by the stakeholders.

– **CQ8:** *What are the relationships among stakeholders?*

- **How can our approach help answering this question?** Using the number of responsibility relations among stakeholders and a list of them.
- **Analysis**: It is possible to know the responsibility between the stakeholders during a process instance execution, detecting whether one stakeholder is responsible for many others or not.
- **Decision-Making Possibility**: after analyzing the responsibility among stakeholders in executed instances, the process manager can use this information when allocating the responsibilities between stakeholders when a new instance of this process model is created, according to the project objectives / goals.

– **CQ9:** *Which roles each stakeholder assumes?*

- **How to answer the question:** Number of roles performed by a stakeholder and a list of them.
- **Analysis**: It is possible to analyze all the roles that have already been played by a specific stakeholder as well as, from a role, to verify which stakeholders can accomplish it.
- **Decision-Making Possibility**: In a next instantiation of this process model, if the process manager needs to allocate some person stakeholder in a specific activity that needs some pre-defined role, he can evaluate who can perform this role, based on stakeholders' skills. On the other hand, he can also decide who should participate in a training programming to be able to accomplish more roles during process execution.

Goal 3: Tracking derivations and revisions among artifacts or procedures

– **CQ10:** *Which artifacts are derivations from others?*

- **How can our approach help answering this question?** Using a graph showing process artifacts and its respective derivation (our approach considers a derivation among two Artifacts when an activity *ac* has used an artifact *art1* and this same activity generates a new artifact *art2*).

- **Analysis**: It is possible to discover all the artifacts derived from others in addition to verify the artifacts that were most used for the derivation of others and, therefore, are of great importance in the analyzed SDP.
- **Decision-Making Possibility**: When verifying that an artifact was much used for the derivation of others, the changes in this artifact must be well planned to avoid that all the various other artifacts derived from it also need to be changed.

– **CQ11:** *Which artifacts or procedures are revisions from others?*

- **How our approach can help in answering this question?** Using a graph showing process artifacts (or procedures) and the revisions relations between them.
- **Analysis**: It is possible to discover all the artifacts revisions, in addition to its latest versions / revisions.
- **Decision-Making Possibility**: It is possible to evaluate when the last revision of a given artifact occurred, in addition to showing if an artifact has already suffered many or no changes. This information can help in defining which artifact (or procedure) can / should be used in a future process execution.

In addition to defining and detailing eleven CQ that the iSPuP approach proposes to answer (using a provenance model and an operational ontology), the relevance of each CQ was evaluated with process managers (Chapter 7), in order to reach the main objective of this thesis (support process analysis and data-driven decision-making).

4.6 Final Remarks

This chapter presented PROV-SwProcess, a provenance model (that extends PROV) to accommodate SPD provenance specificities. In addition to detailing the model itself, a comparison with other provenance models and extensions was made. One of the main characteristics of this model is its ability to infer new information using inference rules (presented in Section 4.4.1). Finally, a series of competency questions that PROV-SwProcess can answer (using an operational ontology) are carefully detailed.

CHAPTER 5 – iSPuP APPROACH

This chapter introduces the iSPuP approach that use PROV-SwProcess as its core model. It supports PROV-SwProcess model instantiation, new information inferencing and data visualization. iSPuP main elements are detailed, as well as its tool support. Finally, a toy example showing this approach in action is presented.

5.1 Introduction

In our vision, the best way to capture the SP provenance data is adapting the process execution engine or the workflow engine[18] used by the organization to collect provenance data. However, most small and medium-sized companies, in the initial levels of software maturity models, do not use such tools to execute their software processes, but rather a set of different tools (e.g., version control system, issue trackers, and documentation management systems). Considering the diversity of such tools, iSPuP (**i**mproving **S**oftware **P**rocess **u**sing **P**rovenance) approach was developed to structure all the recorded execution data according to PROV-SwProcess Model (model instantiation), as well as to allow strategical information discovering (through inferencing mechanisms), besides a module for data visualization and analysis, enabling process managers' data-driven decision-making (BOSCH, 2017). ISPuP approach is detailed in the following subsection.

5.2 iSPuP Phases

Considering PROV-SwProcess model, and this thesis' goal (***Propose and evaluate an approach for capturing, storing, discovering and visualizing SDP execution provenance data to support process analysis and data-driven decision-making***), iSPuP approach is composed by three main phases (Fig. 5.1): (i) Systematics for SDP provenance data capture and storage; (ii) Systematics for deriving SDP implicit information using inference mechanisms; (iii) Systematics for converting SDP provenance data into a graph format aiming to facilitate process manager in a decision making activity. All these phases use as basis the PROV-SwProcess provenance model (presented in Chapter 4). These three main phases have five main activities: (1) Process execution and provenance data capture; (2) Captured data transformation according to

[18] As it is done in cases of scientific workflows.

the PROV-SwProcess model; (3) Data storage and ontology populating; (4) Inference machine execution; and (5) Data visualization and analysis, organized according to the execution flow shown in Figure 5.1.

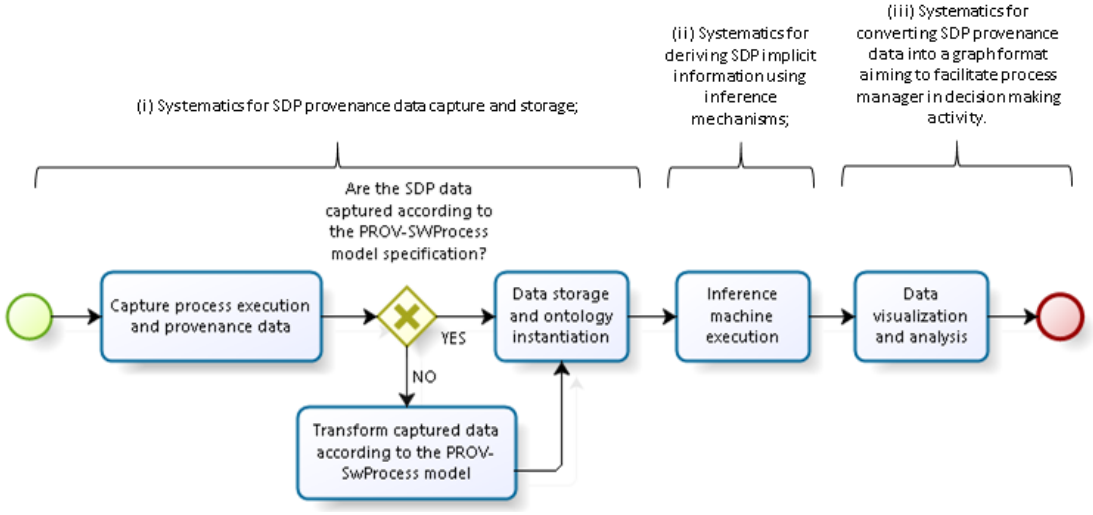


Figure 5.1: Approach Execution Flow.

Considering the first activity, a set of execution data is requested for each of the analyzed processes:

1. Executed processes with their name and responsible (a Stakeholder);
2. Performed activities of each process, with their name, start, and end time;
3. Stakeholders associated with the performed activity (mandatory) and their specific role (optional);
4. Artifacts changed, used, or generated by the performed activity;
5. Procedures adopted for the execution of the performed activity (optional);
6. Hardware and / or Software resources used by the performed activity (optional);
7. Responsibility among stakeholders (optional);
8. Process standard model and process intended model definition, in order to allow process prospective provenance capturing and analysis (all the constructs and relations to define both models are detailed in Figures 4.3. and 4.4, in Chapter 4) (optional).

Although data from items 5 to 8 are optional, it is important to note that to achieve a more accurate and specific data analysis, it is important to record as much data and information as possible. If the data captured in the first activity are not previously organized according to the PROV-SwProcess model, they must be manipulated and organized/stored according to this model. In order to make it possible, a wrapper was specified to make the necessary conversions between different data

formats (to each data format it is necessary a new specific wrapper). Currently, iSPuP approach has three specific wrappers: one for Mantis[19], a wrapper for a proprietary VCS and other that allows converting *.csv* files according to PROV-SwProcess constructs and relations. Besides that, iSPuP tool provides a generic wrapper that needs to be specialized to other specific formats, if it is necessary.

After storing the SDP data in a relational database, modeled according PROV-SwProcess constructs and relations (e.g., we have tables to store activities, artifacts, stakeholders, wasAssociatedWith relation – which relates activities with the stakeholders who have performed them, etc), an ontology is populated (the tool that supports iSPuP approach has a class that makes queries on the relational database and generated individuals in the ontology created according to the model PROV-SwProcess). During activity four, an inference machine - using a reasoner, e.g., Pellet (SIRIN *et al*., 2007) is executed. Lastly, a graph visualization using all the data and new inferred information is generated to allow process manager analysis and support data-driven decision-making. All these activities are presented with more technical details in Section 5.4, which presents the implementation of the architecture created to support iSPuP approach and in Section 5.5, that shows the execution of all these phases using a toy example.

5.3 iSPuP Tool Support

This section details iSPuP tool, developed to allow the previously described phases to be performed. Figure 5.2 shows its architecture with its main layers and elements.

[19] http://www.mantisbt.org/

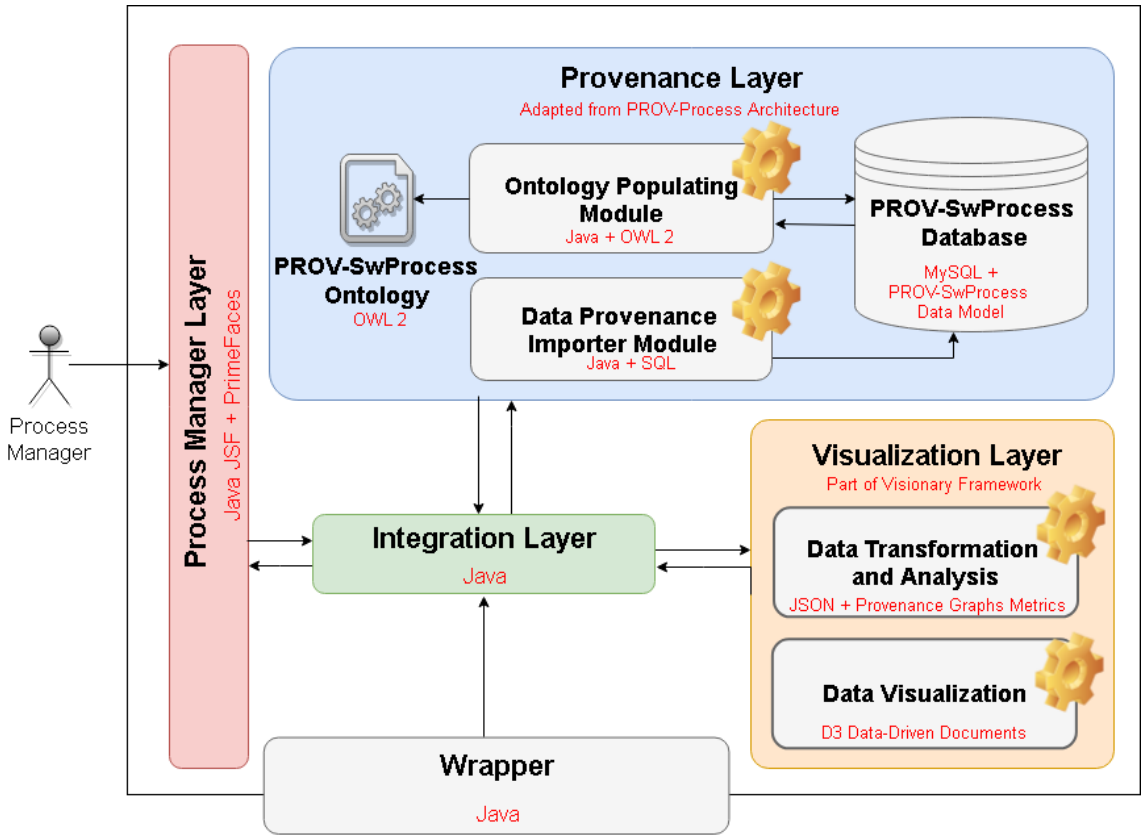


Figure 5.2: iSPuP Tool Architecture.

The iSPuP tool architecture is divided into four distinct layers:

1. **Process Manager Layer**: It is the interface between the process manager and the iSPuP tool and allows interactions / visualizations (produced by the visualization layer) with all process provenance data. It is through this layer that the process manager is able to manipulate SDP execution and provenance data to obtain answers to the competence questions presented in Section 4.5.
2. **Provenance Layer**: this is the main layer of iSPuP architecture. Prospective and retrospective provenance data are captured and stored using these layer's resources, which has a database developed according to PROV-SwProcess data model specification. This layer is also responsible for populating PROV-SwProcess ontology with captured data, enabling inferences into these data using a reasoner and allowing implicit information discovery.
3. **Visualization Layer**: in this layer, SDP provenance and execution data, as well as inferred information, are visually encoded using provenance graphs (e.g., activities and the software process instance are shown in blue rectangles, stakeholders are represented by orange pentagons, artifacts,

procedures, or resources, by yellow ellipses - preserving the PROV notation – and its respective associations are shown using edges) and tables in order to allow process manager analysis and manipulation using these data.

4. **Integration Layer**: this layer is responsible for integrating the other three layers of the approach, to allow the exchange of data/information between them.

In addition to the described layers, the architecture provides a wrapper (briefly described in Section 5.2), to capture the software process definition and the process execution data from a software process management / execution tool.

5.4 iSPuP in Action

The iSPuP tool support was implemented as a web application. Figure 5.2 shows in red the main technologies used to develop each architecture layer.

Process Manager Layer was mainly developed using JavaServer Faces[20] and PrimeFaces[21] in order to allow managers interactions / visualizations (produced by the visualization layer) with of all process execution and provenance data. Examples of the interfaces provided by this layer can be seen in Figures 5.3 and 5.4.

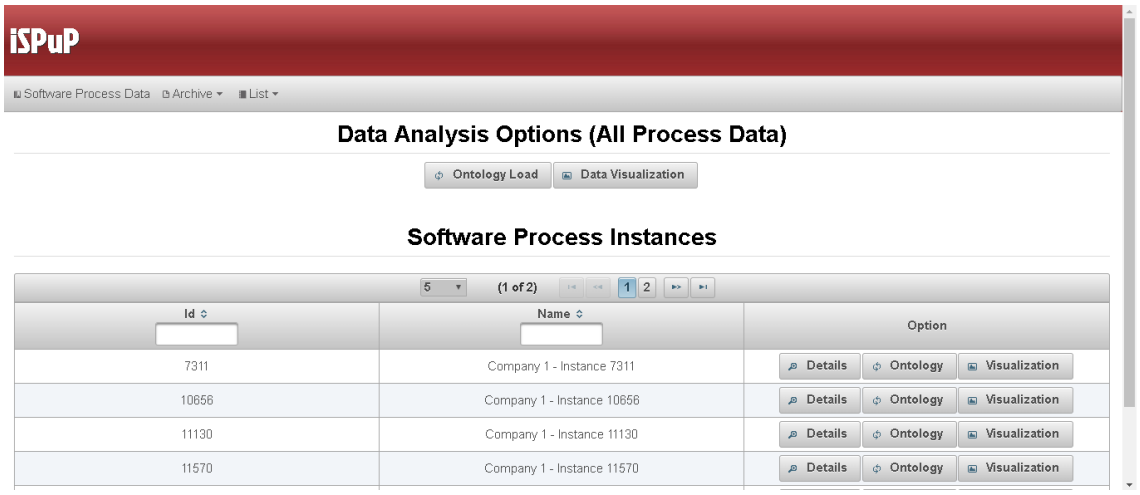


Figure 5.3: Process Manager Interface Example 1.

[20] https://javaee.github.io/javaserverfaces-spec/
[21] https://www.primefaces.org/

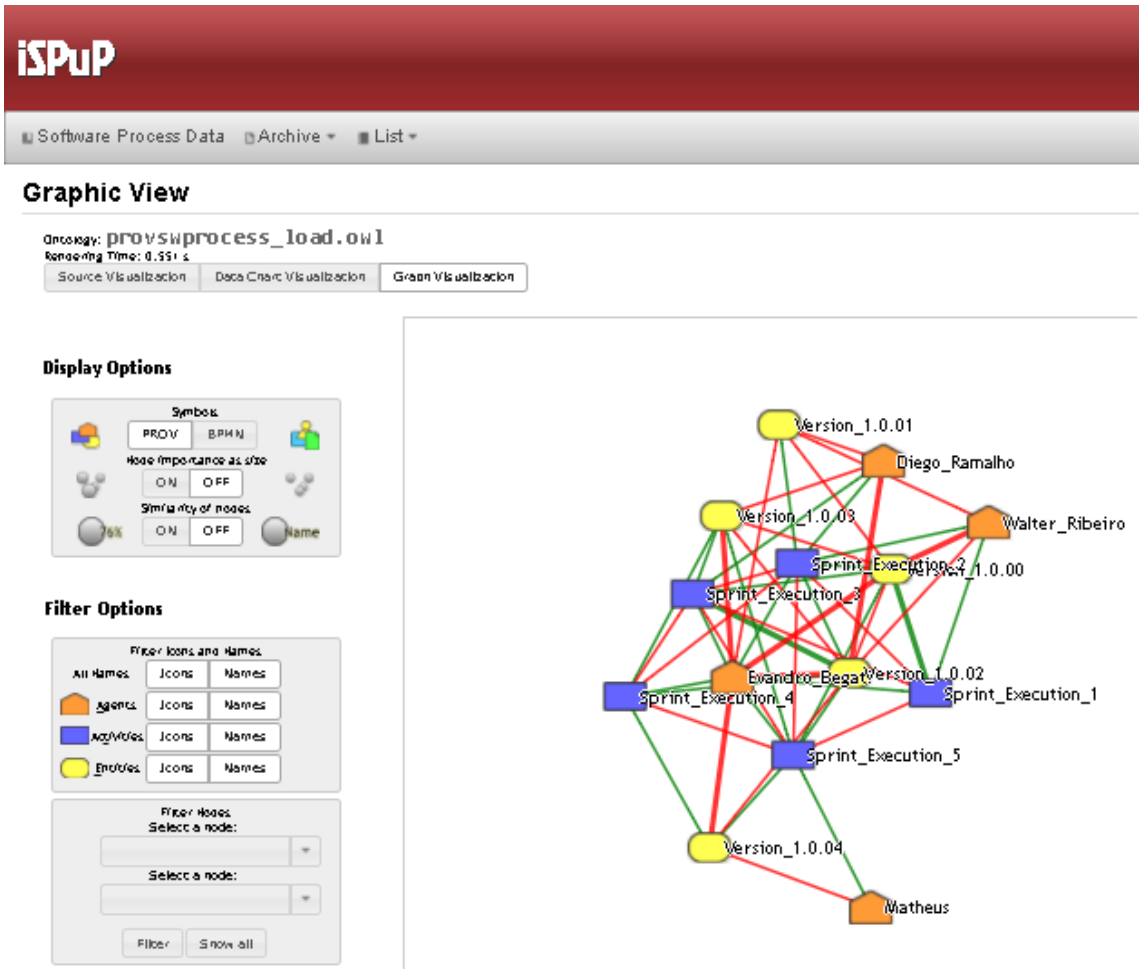


Figure 5.4: Process Manager Interface Example 2.

Provenance Layer has several java classes to allow data import and storing using MySQL SGBD[22], as well PROV-SwProcess ontology population with SDP provenance data (using OWL format).

Visualization Layer transforms the PROV-SwProcess ontology generated in the Provenance Layer into graphs and tables format (encoded in JSON) in order to generate the visualizations to the process manager. Figure 5.4 shows an example of generated graph and Figure 5.5 an example using data table format.

[22] https://www.mysql.com/

Name	Type	Degree	Created Artifacts	Modified Artifacts
Mary	Person_Stakeholder	3	1	0
Simon	Person_Stakeholder	6	0	2
Client	Organization_Stakeholder	2	1	0
Derek	Person_Stakeholder	3	0	1
Support_Team	Team_Stakeholder	2	1	0
Joao	Person_Stakeholder	2	1	0

Figure 5.5: Data Table Example.

In order to present the approach operation, its usefulness, the inference mechanism to derive implicit information and the approach's ability to support SDP analysis and data-driven decision-making with a simple example, a toy example was defined. Using this toy example, we checked if the approach had possible faults or inconsistencies, and some points of improvements in the tool support (e.g., adjustments in the visualizations and tool usability) before its evaluation using real cases / scenarios. The answers to the PROV-SwProcess model competency questions (presented in Subsection 4.5) are also detailed using the toy example data.

Considering the first activity to use the approach (1) *Process execution and provenance data capture*, the data presented in Table 5.1 were created. This toy example is based on a SDP called *New Resource Development*. It is composed by five distinct activities that manipulated specific artifacts, adopted two procedures, and used two resources. Six stakeholders were involved in this SDP.

Table 5.1: Toy Example Execution Data.

Process Instance: New Resource Development Process Responsible Attribution: Simon						
Activities	Start	Ended	Stakeholders	Artifacts	Procedures	Resources
New Resource Specification	2017-01-14 10:00:00	2017-01-15 12:00:00	Client, Joao, Support_Team	**USED** Cliente_Request_Email (Information_Item) **GENERATED** Requirements_Document (Document),		-
Codification	2017-01-15 13:00:00	2017-01-15 18:00:00	Simon (Responsible for Derek), Derek	**USED** Eclipse_IDE (Software_Product), Financial_Module (Software_Item), Requirements_Document (Document), UML_class_model (Model) **CHANGED**	-	-

				Payment_Component (Software_Item)		
Test Cases Definition	2017-01-18 10:00:00	2017-01-18 13:00:00	Mary	**USED** Requirements_Document (Document) **GENERATED** Payment_Test_Cases (Document)	**ADOPTED** Test_Cases_Template (Document_Template)	
Test	2017-01-18 10:00:00	2017-01-18 18:00:00	Mary	**USED** Payment_Component (Software_Item), Payment_Test_Cases (Document)	**ADOPTED** white-box_testing (Technique)	**USED** Dell_Inspiron_ Intel_Core_i7_8GB_1TB (Hardware_Resource), JUnit5 (Software_Resource)
Deploy	2017-01-20 10:00:00	2017-01-21 18:00:00	Simon	**CHANGED** Accounting_System (Software_Product)	-	-

Considering that the proposed approach also addresses SDP prospective provenance, using the process model definition, Table 5.2 presents the data about the toy example process model definition.

Table 5.2: Toy Example Process Model Definition

Process: New Resource Development Process Responsible: Simon				
Activities	Role	Artifacts	Procedures	Resources
New Resource Specification **PRECEDES** Codification	-	**GENERATES** Requirements_Document (Document),	-	-
Codification **PRECEDES** Test	Programmer	**USES** Eclipse_IDE (Software_Product),	-	-
Test Cases Definition **ISSUBACTIVITY** Test	Tester	-	**ADOPTS** Test_Cases_Template (Document_Template)	
Test **PRECEDES** Deploy	Tester	-	-	-
Deploy	-	**CHANGES** Accounting_System (Software_Product)	-	-

The second and the third approach activities were performed (*Captured data transformation according to the PROV-SwProcess model*; *Data storage and ontology generation*) and the obtained visual result is shown in Figure 5.6 (in this example we chose to show the visualization generated by the tool before the execution of the inference machine – the fourth approach activity – in order to show the differences between the proposed visualization without and with the inferences). Activities and the software process instance are shown in blue rectangles, stakeholders are the orange pentagons and artifacts, procedures, or resources, in yellow ellipses (preserving the PROV notation). The relations between them are shown as green edges.

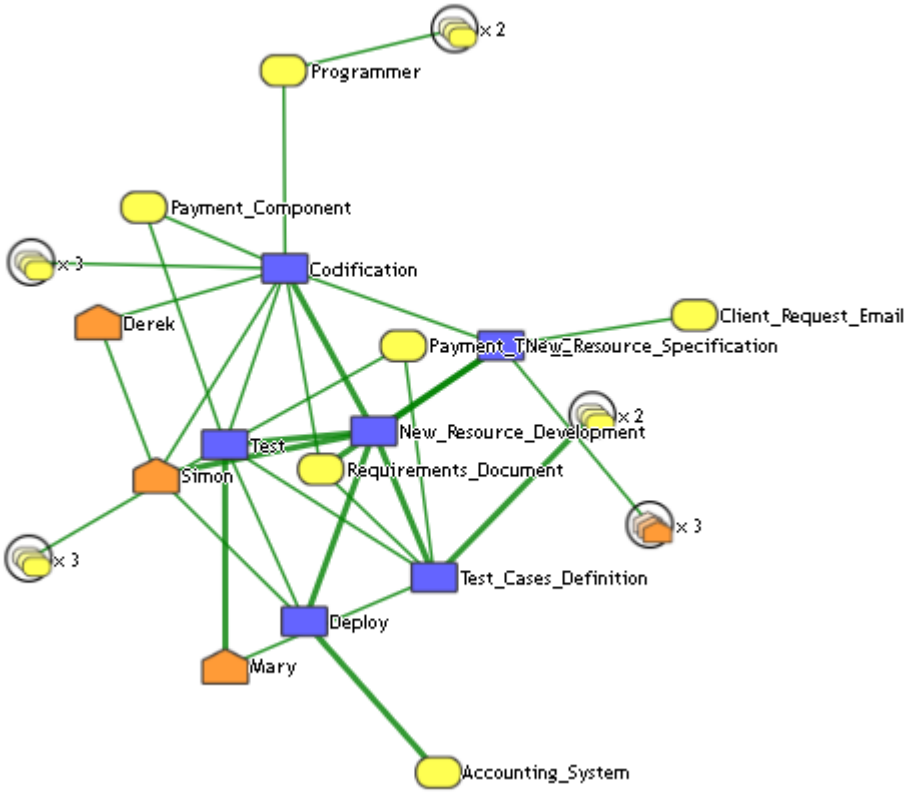


Figure 5.6: Toy Example Without Inferences.

Considering that the approach uses Visionary Framework (OLIVEIRA *et al.*, 2017) to support the visualization of the data, this framework automatically groups similar nodes. This grouping happened with the stakeholders *Client*, *Support_Team*, and *Joao* (Figure 5.7), to facilitate and simplify the visualization of the generated provenance graph. In the presented example, both stakeholders had only the *wasAssociatedWith* relation with the activity *New_Resource_Specification* (Figure 5.8) and, for this reason, they were considered similar.

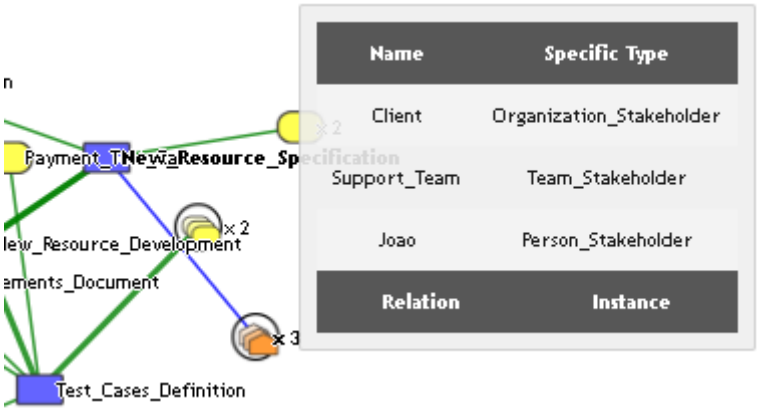


Figure 5.7: Toy Example – Stakeholders Grouping Members.

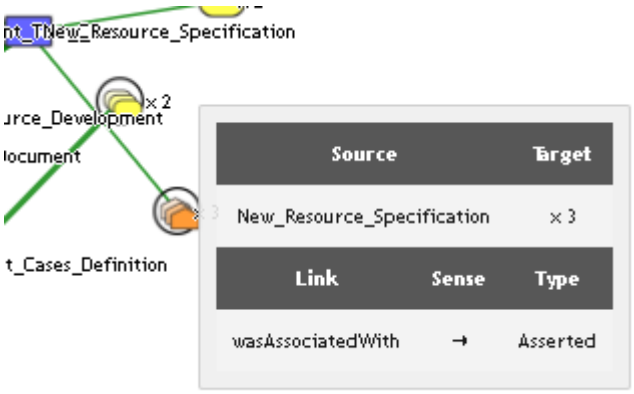


Figure 5.8: Toy Example – Stakeholders Grouping Association.

When the fourth and fifth activities were performed (*Inference machine execution;* and *Data visualization and analysis*), we obtained the visualization shown in Figure 5.9. All the implicit relations proposed by the PROV-SwProcess model were inferred (only the *WasComposedBy* inference does not appear, because this inference is only used when more than one process instance is being analyzed) and appeared in red in Figure 5.9.

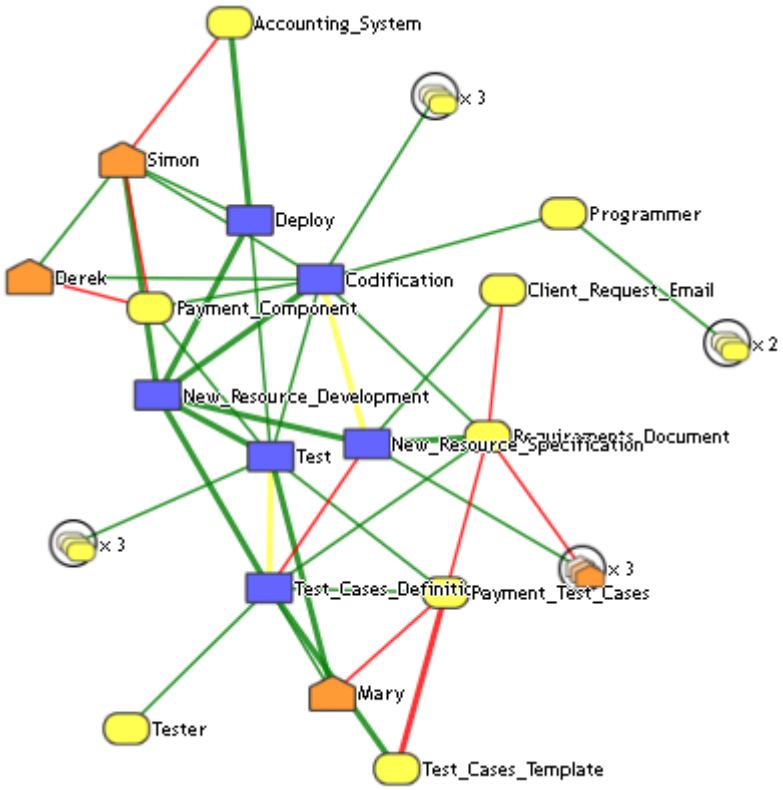


Figure 5.9: Toy Example with Inferences.

In addition to the graphical representation, the approach tool also presents some data in a tabular format (Figure 5.10), in order to facilitate their analysis and to reach the eleven specific goals. Figure 5.10 shows, for example, a zoom on this table, after applying a filter to display only the stakeholders involved in this process instance.

Name	Type	Degree	Created Artifacts	Modified Artifacts
Mary	Person_Stakeholder	3	1	0
Simon	Person_Stakeholder	6	0	2
Client	Organization_Stakeholder	2	1	0
Derek	Person_Stakeholder	3	0	1
Support_Team	Team_Stakeholder	2	1	0
Joao	Person_Stakeholder	2	1	0

Figure 5.10: Toy Example - Data Analysis Table.

After the generation of the visualizations previously presented, the process manager can perform the analysis of these data.

Next, we explain the views that support the achievement of the eleven specific objectives of the approach, using the toy example.

Goal 1: Process structure identification during execution and possibilities for process redesign

- **CQ1** *What are the process activities, artifacts, resources, procedures, stakeholders, and the relations among them during the process execution?*

Figure 5.11 shows all the process elements from the toy example and their relations during the process execution. When hovering the mouse on each node or relation, we can see its name and details. Several analyses can be done using this visualization. For examples: we can see that *Simon* and *Derek* participated in the *Codification* activity, but only *Simon* acted on *Deploy* activity; *Mary* acted as a *Tester* and created the document *Payment_Test_Cases*. Using this visualization, if any gap (elements without association or inadequate relation established) was found, the process manager can use this information to correct it in next process executions.

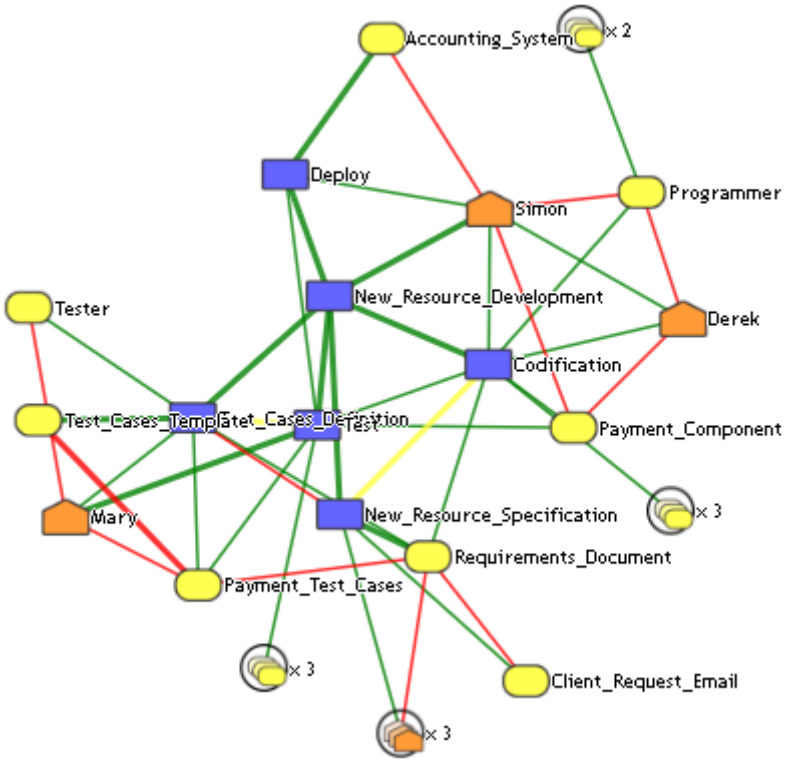
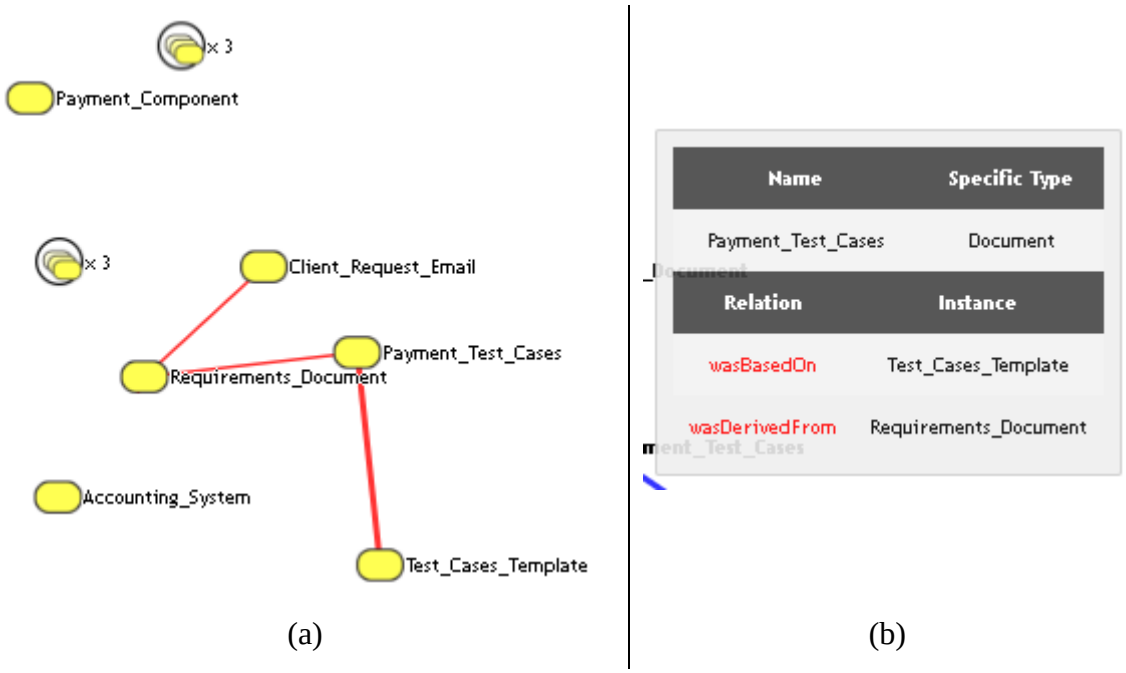


Figure 5.11: Toy Example - Visualization to support CQ1.

— **CQ2:** *Which procedures are used by the process during its execution?*

Figure 5.12(a) shows the process artifacts, procedures and resources and its direct relations. Figure 5.12(b) shows the popup that is displayed when hovering the mouse in *Payment_Test_Cases*. Based on these visualizations, we can see that a procedure called *Test_Cases_Template* was necessary to create the document artifact *Payment_Test_Cases*, and only the document artifact *Payment_Test_Cases* was based on the *Test_Cases_Template*. As a data-driven decision-making possibility, in the case of this toy example, most of the artifacts were not created based on any procedure, so this fact could be analyzed with the purpose of establishing specific procedures for the creation of the different types of artifacts involved in the process.



(a) (b)

Figure 5.12: Toy Example - Visualization to support CQ2.

— **CQ3:** *Which activities had a high complexity (considering the number of associated stakeholders, artifacts, procedures and / or resources)?*

Figure 5.11 (that shows all the process elements) can be firstly used to support this question. Considering this visualization, we cannot see any discrepancy between the activities associations to detect any complex activity. We can only detect that *Deploy* activity has fewer associations than the other activities. Another way to support in answering this question is using the tabular view (Figure 5.13), using the *Activity Degree* information. As it was detected in the graphic visualization, the activity with the lowest associations is *Deploy* and the rest has between 8 and 11 relations.

id	Name	Type	Degree
13	Deploy	Activity	4
0	Test_Cases_Definition	Activity	8
20	New_Resource_Specification	Activity	8
6	Test	Activity	10
15	Codification	Activity	11

Figure 5.13: Toy Example - Visualization to support CQ3.

– **CQ4:** *Which activities had a high dependency (on other activities)?*

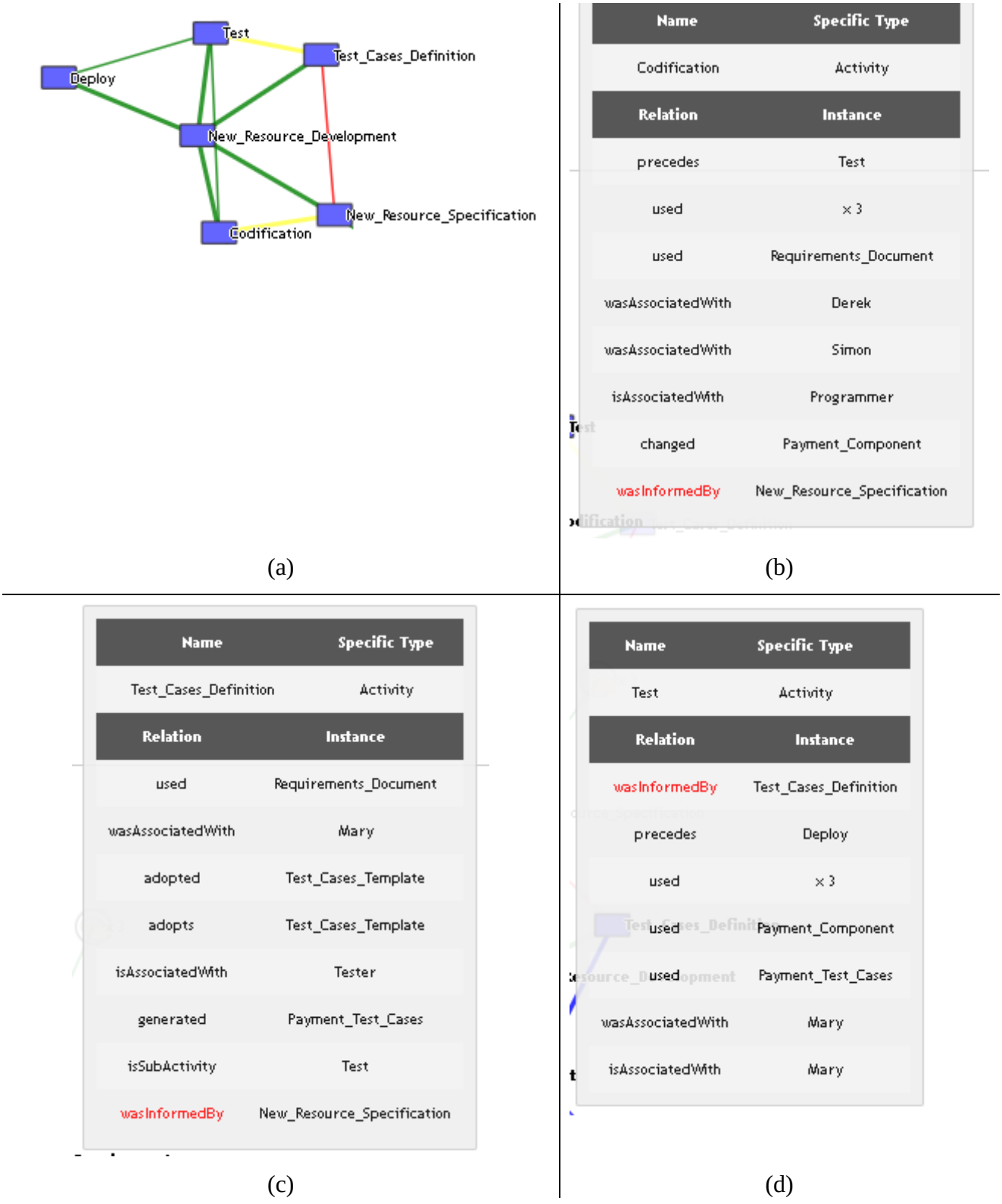


Figure 5.14: Toy Example – Visualization to support CQ4.

Figure 5.14(a) shows the process instance (*New_Resource_Development*) and its activities (*Deploy*, *Test*, *Test_Cases_Definition*, *Codification*, and

New_Resource_Specification) - a filter was used in the visualization tool only to show them). Figure 5.14 (b), (c) and (d) shows the details of *Codification*, *Test_Cases_Definition*, and *Test* activities respectively. *Codification* and *Test_Cases_Definition* depends on the *New_Resource_Specification* and *Test* depends on the *Test_Cases_Definition*. Then, it was not verified in the toy example any relevant difference between the activities dependency during its execution flow and its respective process model flow (in this toy example, the process flow model is continuous, following the same order in which the activities are presented in Table 5.1).

Goal 2: Understanding stakeholder's involvement in process execution

– **CQ5:** *What is the activities distribution among stakeholders?*

Figure 5.15 shows the process instance stakeholders, their associated activities and software_process (a filter was used in the visualization tool to omit the artifacts, procedures, and resources). Another way to support answering CQ5 is by using part of the data presented in the approach data table, as can be seen in Figure 5.16. A filter was used in this table to show only the stakeholders.

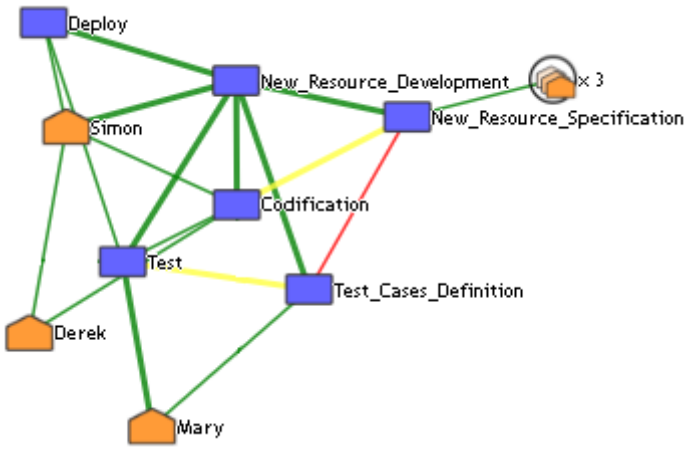


Figure 5.15: Toy Example - Visualization to support CQ5 – part 1.

Questions like "*In how many activities a stakeholder participated? / Which stakeholder participated in more activities? / Which stakeholder participated in fewer activities?*" can be easily answered using the visualization presented in Figure 5.15 or in the table shown in Figure 5.16. We cannot perceive any great discrepancy between the number of stakeholders associated activities. While *Mary* and *Simon* were associated with two activities, other stakeholders are associated with only one activity. However, if it were verified that a stakeholder is participating in much more activities than others,

the process manager could evaluate if this fact was really planned/expected or if it has been occurring due to an inadequate activity distribution during the process instantiation.

Name	Type	Degree	Created Artifacts	Modified Artifacts	Associated Activities
Mary	Person_Stakeholder	3	1	0	2
Simon	Person_Stakeholder	6	0	2	2
Client	Organization_Stakeholder	2	1	0	1
Derek	Person_Stakeholder	3	0	1	1
Support_Team	Team_Stakeholder	2	1	0	1
Joao	Person_Stakeholder	2	1	0	1

Figure 5.16: Toy Example - Visualization to support CQ5 – part 2.

– **CQ6:** *Which artifacts are known by a stakeholder, considering that in some process execution he/she created or modified such artifact?*

Figure 5.17 shows the process artifacts and stakeholders. Using this visualization, we can see that the group of three stakeholders (*Client*, *Support_Team*, and *Joao*) knows about *Requirements_Document*, *Mary* knows the *Payment_Test_Cases*, *Simon* and *Derek* have knowledge about *Payment_Component*, and *Simon* also knows the *Accounting_System.*

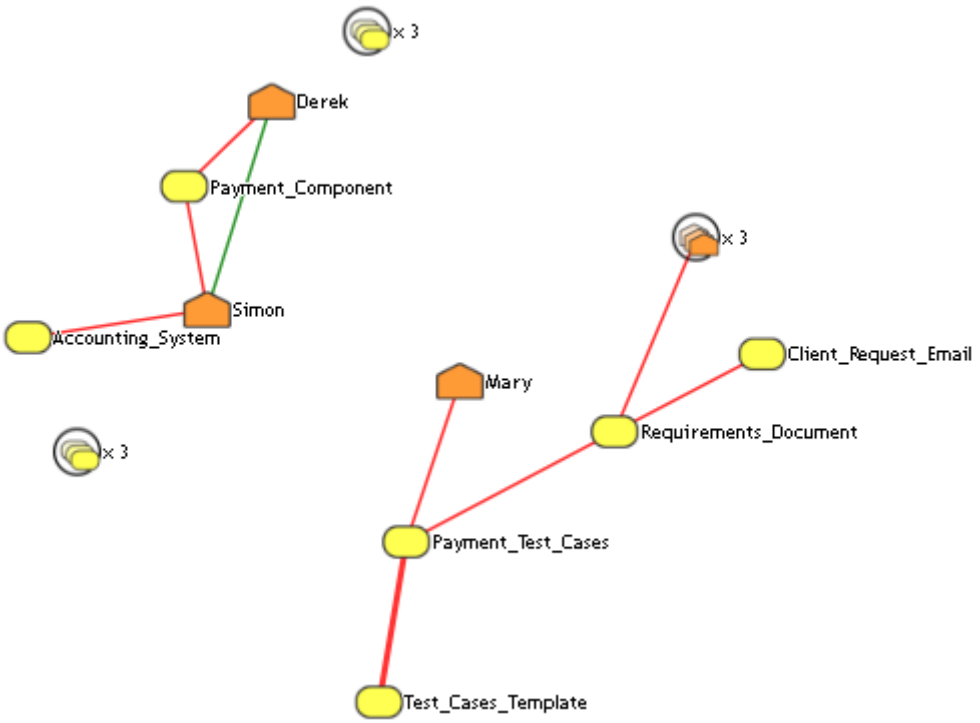


Figure 5.17: Toy Example - Visualization to support CQ6.

We can also detect that there is no great discrepancy among the stakeholders, considering the number of artifacts that they have knowledge of. All the stakeholders are associated to just one or two artifacts, creating or modifying it (*Payment_Component* and *Accounting_System* were modified while *Requirements_Document* and *Payment_Test_Cases* were created – the relation name can be visualized in the tool when hovering the mouse over it). In a future execution of the analyzed process, if a certain task is associated with the *Payment_Component,* the process manager can allocate *Simon* or *Derek* to this task, considering that they have a previous knowledge in this artifact.

– **CQ7:** *Which stakeholders are out of the average of created and/or modified artifacts?*

To support in answering CQ7, we use part of the data presented in the approach data table, as can be seen in Figure 5.18. A filter was used in this table to show only the stakeholders.

Name	Type	Degree	Created Artifacts	Modified Artifacts
Mary	Person_Stakeholder	3	1	0
Simon	Person_Stakeholder	6	0	2
Client	Organization_Stakeholder	2	1	0
Derek	Person_Stakeholder	3	0	1
Support_Team	Team_Stakeholder	2	1	0
Joao	Person_Stakeholder	2	1	0

Figure 5.18: Toy Example - Visualization to support CQ7.

Considering the toy example, we cannot observe any great discrepancy between the number of artifacts created or modified by the stakeholders. While *Mary*, *Client*, *Support_Team,* and *Joao* created one artifact each, *Simon* modified two artifacts and *Derek* only one. However, if it were verified that a stakeholder created much more artifacts than others, the process manager should evaluate this fact to understand it and check if they really need to be created or if there is a stakeholder's lack of knowledge about the existing artifacts. Another possibility for decision-making would be to

evaluate in a next execution of the process example the possibility of allowing the stakeholders who created artifacts to modify existing artifacts and those who only modified artifacts the possibility to create new ones, if necessary.

– **CQ8:** *What are the relationships among stakeholders?*

Figure 5.19 shows the stakeholders and their direct relations (a filter was used in the visualization tool to show only the stakeholders).

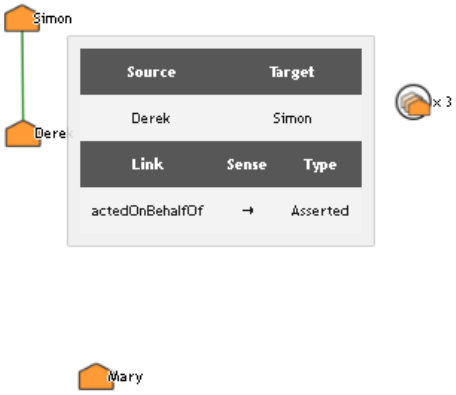


Figure 5.19: Toy Example - Visualization to support CQ8.

According to Figure 5.19, only *Derek* acted on behalf of *Simon*. Considering this question is supported by the PROV-SwProcess association called *ActedOnBehalfOf* and it is not inferred, the visualization only shows the information provided in the process execution data. Considering that there is only one relationship of responsibility among stakeholders, the process manager should assess if the other stakeholders do not really act under the responsibility of others.

– **CQ9:** *Which roles does each stakeholder assume?*

Figure 5.20 shows the stakeholders and its performed roles. Considering that the Visionary Framework and PROV notation do not have a specific symbol for the stakeholder's roles, we assume that this visualization should be improved in order to facilitate process manager interpretation (future work). According to this visualization, *Derek* and *Simon* acted as Programmers and Mary acted as a *Tester*. As a data-driven decision-making example, we can state that in a next execution of the process, if the process manager needs to allocate a *Tester* or a *Programmer* in a specific activity, he/she knows who can perform these roles, based on previous execution data.

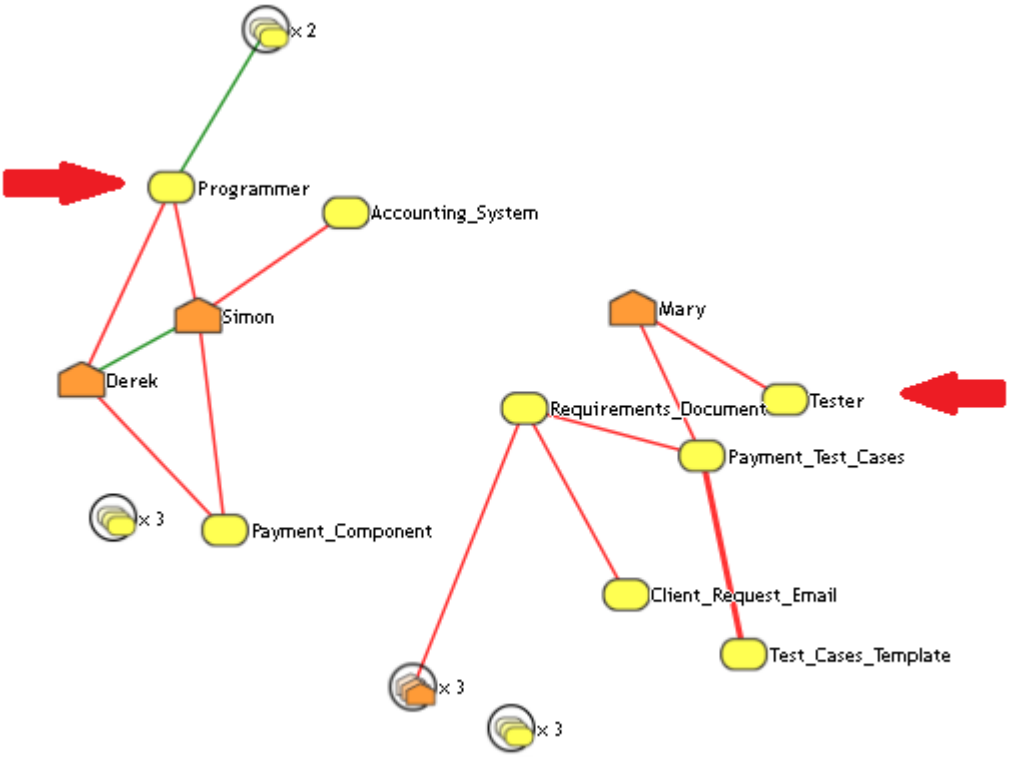


Figure 5.20: Toy Example - Visualization to support CQ9.

Goal 3: Tracking derivations and revisions among artifacts or procedures

– **CQ10:** *Which artifacts are derivations from others?* and

– **CQ11:** *Which artifacts or procedures are revisions from others?*

Figure 5.21 shows only the process artifacts and procedures and its direct relations in order to support in answering CQ10 and CQ11. When hovering the mouse in *Payment_Test_Cases* and *Requirements_Documents* we can see Figure 5.22 (a) and (b). Using these visualizations, we can see that *Requirements_Document* was derived from *Cliente_Request_Email* and *Payment_Test_Cases* was derived from *Requirements_Document*. Besides that, if an artifact was a revision from other(s), the tooltip will detail this information. It was not verified in the toy example an artifact that was much used for the derivation of others or had many revisions. If it were verified that an artifact was much used for the derivation of others, the changes in this artifact would have to be well planned to avoid that all the various other artifacts derived from it also need to be changed.

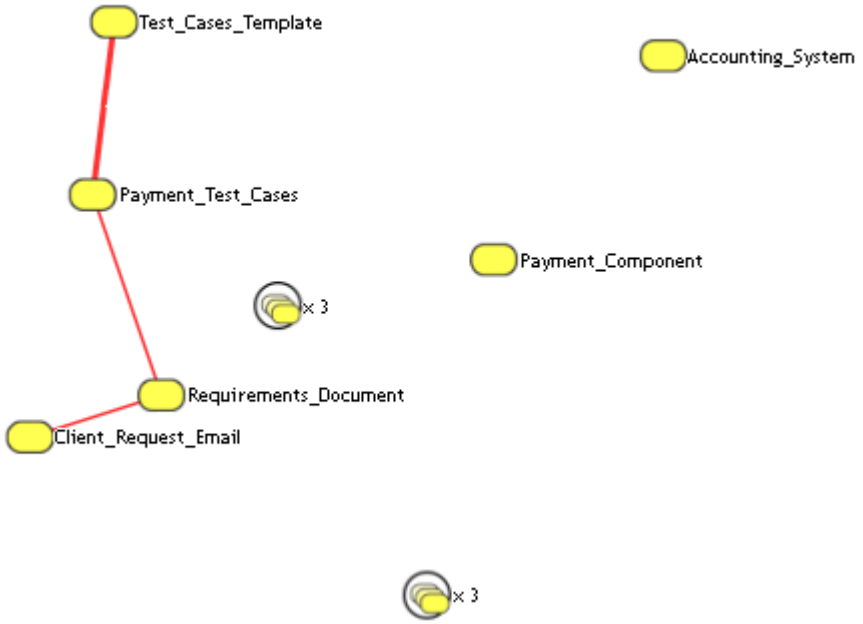
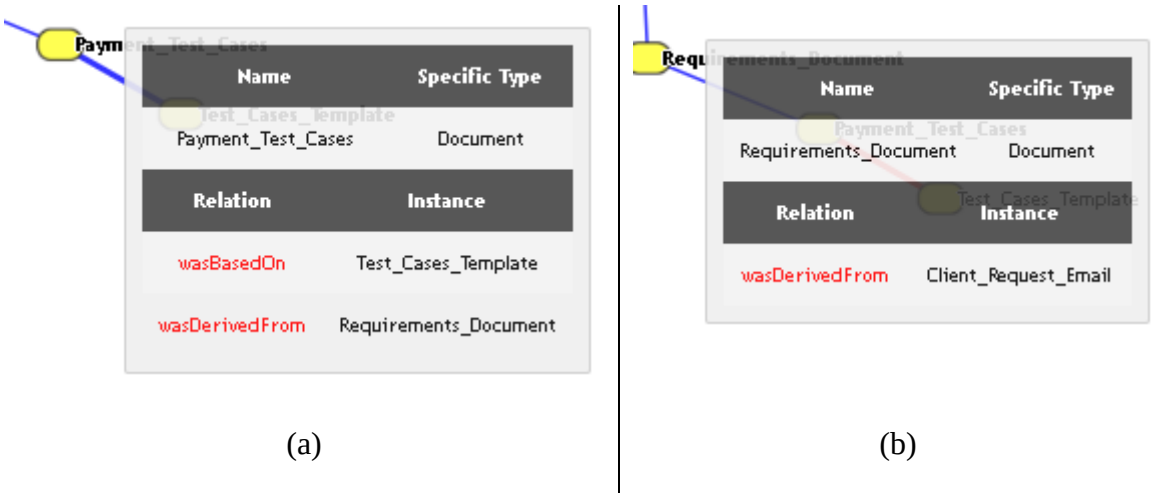


Figure 5.21: Toy Example - Visualization to support CQ10 and CQ11 – part 1.



(a) (b)

Figure 5.22: Toy Example - Visualization to support CQ10 and CQ11 – part 2.

5.5 Final Remarks

This chapter presented the iSPuP approach, its main phases and architecture elements. The core of this approach is its provenance model, PROV-SwProcess, that was carefully detailed in Chapter 4. The proposed approach can be used in answering eleven competency question, in order to achieve three specific goals (1 - Process Structure Identification and possibilities for process redesign; 2 - Understanding stakeholder's involvement in process execution; and 3 - Tracking derivations and

revisions among artifacts or procedures). A toy example was used to explain the operation of iSPuP tool with simple examples, showing how to answer each competency questions using the visualizations provided by the tool support.

The initial effort required to use the approach is the development of a wrapper, to structure all the process recorded execution data according to PROV-SwProcess Model, besides some training about the tool support. However, the process should not be changed to use our approach and it could be used to any kind of SDP.

Next chapter presents the experiments that were conducted to evaluate both PROV-SwProcess model (with provenance and process experts) and the approach as a whole, using three distinct processes from industry, in order to verify iSPuP's ability in supporting SDP analysis and data-driven decision-making in real SDP contexts.

CHAPTER 6 – PROV-SwProcess EVALUATION

This chapter describes PROV-SwProcess Model evaluation, *which is an inspection of PROV-SwProcess Model with provenance and process experts.*

6.1 Introduction

PROV-SwProcess model was developed using PROV (MOREAU and GROTH, 2013) and Software Process Ontology (SPO) (FALBO and BERTOLLO, 2009) as basis. These two models have already been extensively tested and validated in their respective areas (provenance and processes) (MISSIER *et al*., 2013b), (PIMENTEL *et al*., 2018), (BHATIA *et al*., 2016), (RUY *et al*., 2016). This fact was one of the reasons for choosing them as the basis for PROV-SwProcess, aiming to encompass the SDP execution and provenance data that should be captured (addressing our first Research Question: ***RQ1***. *What SDP execution and provenance data should be captured?*, presented in Chapter 1).

After defining the data that should be captured and their respective relations, we did a careful analysis about what information could be inferred, using inference rules previously established, what implicit information could be derived from process execution and provenance data. We made this effort aiming to answer ***RQ2***. *Which implicit information can be derived from captured data?*.

Even using two reference models for PROV-SwProcess elaboration, it was decided to evaluate our model by process and provenance experts, in order to ensure its correctness. Considering this fact, PROV-SwProcess model evaluation was planned as a model inspection with experts in provenance and software process.

6.2 Materials and Method

In this evaluation, we used a specific questionnaire to support the detection of possible semantic defects and improvements points in PROV-SwProcess model.

Differently from syntactic defects, which can be easily detected with a tool support, semantic defects are dependent on contextual interpretation and human judgement (DE MELLO *et al*., 2016). Considering this fact, this evaluation was planned to be performed with the help of experts in the domain.

The proposed model inspection was done through a questionnaire, elaborated as a list of possible Discrepant Cases (DCs) to be analyzed by the subjects. DCs are issues suggesting defects or general situations in which defects can be detected (SHULL *et al.*, 2000) and make explicit for the subjects the perspectives to look for defects.

The definition of DCs to compose the questionnaire intended to cover all the PROV-SwProcess constructs and follows the defect taxonomy presented in Table 6.1.

Table 6.1. Defects taxonomy (adapted from [TEIXEIRA *et al.*, 2015]).

Defect category	Description
Omission	The construct omits necessary information about the provenance of software development process
Incorrect fact	Information in the construct contradicts the provenance description or general knowledge about software development process
Inconsistency	Information in a certain part of the construct is not consistent with information in another part
Ambiguity	Information is not clear, allowing multiple interpretations
Extraneous information	Information in the construct is out of scope

PROV-SwProcess is divided into associations (or relations), classes and specific inferences rules. In this vein, DCs are elaborated for all these constructs. As an example, the following DCs were formulated to evaluate activities associations:

- **Omission**: Some association needed to describe the activities that were performed in a software development process (in addition to ***wasAssociatedWith***, ***hadSubActivity***, ***wasInformedBy***, ***adopted***, ***changed***, ***used***, ***startedAtTime***, ***endedAtTime***) was omitted from the model; and some association needed to describe the activities to be executed in a software development process (in addition to ***precedes***, ***dependsOn***, ***hasSubActivity***) was omitted from the model.
- **Incorrect fact**: Some activity association is not compliant with software development process.
- **Inconsistency**: Some activity association has the same semantic meaning (is duplicated in the model).

- **Ambiguity**: Some activity association is not clearly described, using ambiguous terms.
- **Extraneous information**: Some activity association does not belong to the provenance of software development process.

All the elaborated DCs for the constructs (associations, classes, and inference rules) are presented in APPENDIX C and the questionnaire was created considering these DCs (the complete questionnaire is in APPENDIX E, for the first version of PROV-SwProcess model, and in APPENDIX F, for its second version). For each of the questions, we indicate that one of the following items must be chosen: *Yes*; *I don't know / I am not sure* and *No*. *Yes* as an answer means that the expert has found some semantic defect in the model. In these cases, we would like to receive some explanation. Then, based on this explanation, some change in PROV-SwProcess could be evaluated, trying to solve the defect. When the expert answers *No*, it means that the element in evaluation has no semantic defect. *I don't know / I am not sure* was applied when the expert had doubts about some specific element. As an example, the following question was created to analyze some **Omission** in the model inference rules:

F-Q1) *Is some inference rule needed to describe the provenance of software development process omitted from the model?*

*[ ] Yes – Justification:*_____

[ ] I don't know / I am not sure

[ ] No

Before answering the questionnaire, the subject should complete a characterization form (APPENDIX D) and read PROV-SwProcess model specification[23]. Both the characterization form and the questionnaire were developed to be self-administrated by the subjects (an e-mail with instructions was sent to the subjects with the instructions and they fill in them, without any help).

Considering the subject selections, they were chosen based on their expertise in the model area (software process and provenance).

[23] As an example, the last version of PROV-SwProcess specification are available at: http://gabriellacastro.com.br/provswprocess/v3.html

6.3 Results and Discussion

PROV-SwProcess model presented in this thesis is in its third version. It was generated after two rounds of evaluation with experts in software process and provenance.

In the first round, two experts in software process and provenance evaluated the first version of PROV-SwProcess model[24] and the answered questionnaire had 32 questions (APPENDIX E).

Subject 1 has 10 years of academic experience, good knowledge in software process, having studied about this topic in a course/discipline, by reading one or more books, uses his knowledge about this topic in the context of a course in practice, acts as a software engineering analyst, modeling and describing systems, and has a superficial knowledge about provenance.

Subject 2 has 11 years of academic experience, superficial knowledge in software process, however, he has a good knowledge about provenance, uses his knowledge about this topic in the context of a course in practice and in industry projects. He also has a good knowledge about provenance models and PROV.

During the first round, *Subject 1* found 9 defects (out of 32 DCs) and presented 2 uncertainties, while *Subject 2* found only 1 defect. Analyzing these numbers (Figure 6.1), it is possible to note that the percentage of defects found was much lower than the number of correct elements in the model (81% of correct items versus 16% of defects and 3% of uncertainties). Figure 6.2 considers the defects' types. 70% of them are about some model omission (e.g., considering stakeholders associations, *Subject 1* asked: "*How is it possible to define the role performed by the stakeholder?"*), 20% are about some inconsistence (e.g., "*Precedes/ Depends on? These two associations seem to have similar semantic meaning…*") and 10% cites an ambiguity (e.g., "*wasInformedBy – Its meaning is not clear*"). After receiving the questionnaire answers, a direct conversation with the experts was conducted to understand the expert's reasoning and what could be done in the model to eliminate the errors and uncertainties found. After that, as significant changes were made in the model, we considered the need for a re-evaluation of PROV-SwProcess model after this first evaluation round.

[24] Available at: http://www.gabriellacastro.com.br/provswprocess/v1.html

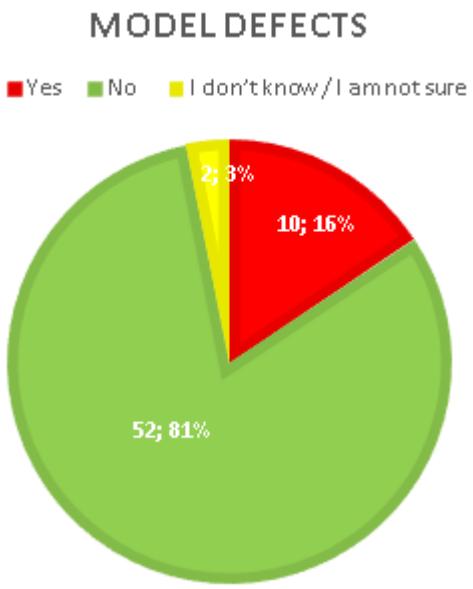


Figure 6.1: Evaluation with Experts – First Round – Model Defects.

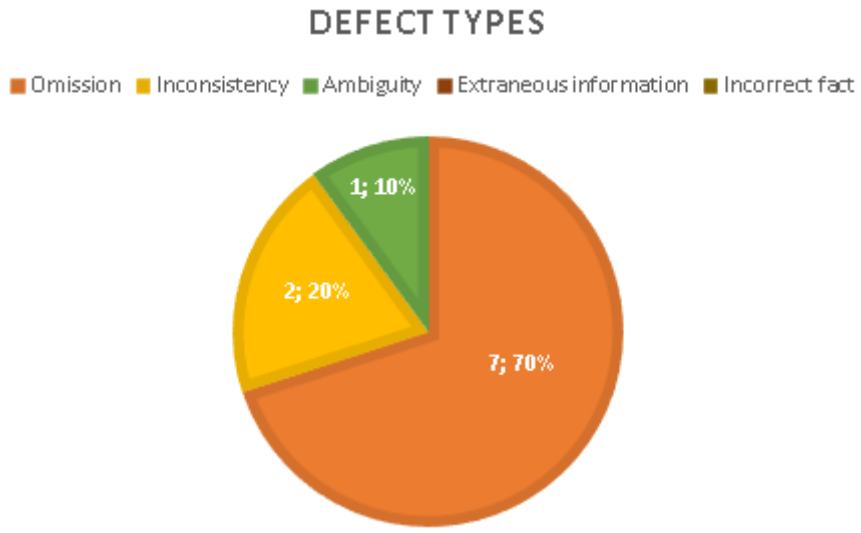


Figure 6.2: Evaluation with Experts – First Round – Defects Types.

The second round follows the same format of the first, with another expert, having a PhD degree, 10 years of academic experience, good knowledge about software process (industry projects), good knowledge about provenance, provenance models, and PROV, and used his knowledge of all the aforementioned topics in academic projects and researches. This round considers the second version of PROV-SwProcess[25]. Some adjustments were made to the form to accommodate the model first version corrections,

[25] Available at: http://www.gabriellacastro.com.br/provswprocess/v2.html

e.g., new added relations/concepts. This evaluation form has 38 questions and the expert pointed out 32 correct points and 6 defects (3 incorrect facts, 1 inconsistence and 2 omissions), listed in the following, with the respective corrections/alterations in PROV-SwProcess model:

- **Incorrect Facts**
    1. A-Q3) Is some software process association not compliant with software development process? *Yes – Justification: I was wondering why HasResponsible relates to a specific stakeholder rather than a role. You may consider three levels of process: defined (with roles only), instantiated (with people assigned), and executed (retrospective info).*

       Model Adjustment: PROV-SwProcess Prospective level was divided in two distinct levels: (i) Process Level (with *HasResponsibleRole* and *IsAssociatedWithRole* new associations), and (ii) Instantiated Level (with *Process_Instance* as a new class and a new association named *IsInstanceOf*).
    2. B-Q3) Is some activity association not compliant with software development process? *Yes – Justification: Again, you may consider having activities defined in different abstraction levels: just connected to role, role assigned to a specific stakeholder (prospective) and stakeholder that actually executed the activity (retrospective). The same reasoning may apply to Artifacts and Resources. When defining an activity, one can specify that a "room" is needed. When instantiating the activity, we bind a concrete room (room 304, for example) to the activity. However, during the execution, another room (room 443, for example) may be used. The same for Artifacts: when defining the activity I know it produces "code". However, in the retrospective view of the activity, I know that it produced class1.java, class2.java, etc.*

       Model Adjustment: The changes made in the previous item already solve the mentioned defect.
    3. C-Q3) Is the stakeholder association not compliant with software development process? *Yes – Justification: I was wondering if you could use the Composite pattern on "Team Stakeholder" to allow a precise definition of the stakeholders that belong to the team, or even teams that belong to the team (necessary for Scrum of Scrums, for instance).*

Model Alteration: The association *Participates* was created in the Instantiated Level.

- **Inconsistency**
    1. F-Q3) Has some class the same semantic meaning (is duplicated in the model)? *Yes – Justification: Model seems to be a type of Document.* Model Adjustment: Considering SPO ontology (the ontology on which PROV-SwProcess model was based), a *Model* is not a type of *Document* - they are *disjoint concepts*. Based on this and trying to solve this pointed defect, the definition of *Document* was rewritten, in order to make the semantics of a document clearer in our model.
- **Omissions**
    1. D-Q1) Is some occurred association whose origin is a software process artifact (in addition to wasBasedOn, wasDerivedFrom, generatedAtTime, and invalidatedAtTime) omitted from the model? *Yes – Justification: To fix the problem of definition (e.g., code) vs instance (e.g., class1.java) that I pointed in B-Q3, you may consider adding an instanceOf association. Another possible association is revisionOf, to clearly identify versions of the same instance (v1 and v2 of class1.java). You may also use wasDerivedFrom for this, adding a property to indicate the type of derivation (revision).*

       Model Adjustment: the creation of two levels of prospective provenance solves part of what was presented in the justification and the association *WasRevisionOf* relationship was also created, at the Retrospective Provenance level, for both artifacts and procedures, as suggested.
    2. E-Q1) Is some occurred association whose origin is a procedure (in addition to wasAppliedTo) omitted from the model? *Yes – Justification: considering that procedures also evolve over time, you may want to track such evolution with revisionOf or wasDerivedFrom, as I discussed before*.

       Model Adjustment: The changes made in the previous item already solve the mentioned defect.

At the end of the questionnaire, there was a space for g*eneral comments about the model* and g*eneral comments about the evaluation*. For the first, the expert wrote

"*Very nice piece of work!*" and commented that the evaluation "*Took more time than anticipated but was worth it.*"

Although a new analysis of this third version was not performed by a fourth expert, we chose to evaluate this last version through an instantiation of the model with real data, as will be presented in the next chapter.

6.4 Threats to Validity

Despite our care in reducing the threats to validity of the evaluation with experts, there are some factors that can influence the obtained results. The subjects' selection can affect the results because of the natural variation in human reasoning / knowledge and considering that there are no wrong or right answers in the experiment. However, the evaluation was executed with experts on a voluntary way, considering that volunteers are more motivated for executing tasks. Besides that, the subjects were defined according to their knowledge in the approach related areas (SDP and provenance). In addition, the expert's evaluation was performed offline, without any follow-up from the researcher.

6.5 Final Remarks

Considering the main problem analyzed in this thesis (*How to capture and analyze what really occurred during a software development process execution in order to support process analysis and data-driven decision-making?)*, PROV-SwProcess model was developed. This chapter presents the conducted model evaluation with provenance and process experts. Two rounds of PROV-SwProcess model inspection were conducted (the first with two experts and the second with a different expert). The founded defects pointed by the experts are detailed with a discussion of how they were corrected.

CHAPTER 7 – iSPuP EVALUATION

This chapter describes iSPuP approach evaluation, which uses SDP data from three distinct companies and includes an interview with its managers.

7.1 Introduction

Aiming to answer the last two research questions presented in Chapter 1 (***RQ4***. *What are the analysis possibilities that can be carried out on the captured data?* and ***RQ5***. *How SDP analysis can help in process manager decision-making?*) [26], we want to investigate them in real industry scenarios. We are interested in evaluating the iSPuP approach feasibility, using PROV-SwProcess provenance model, in real world contexts.

Case study is a standard method used for empirical studies in several sciences and is well suited for the industrial evaluation of software engineering methods and tools (WOHLIN *et al*. 2012) (YIN, 2014). In this vein, a case study was the most suitable choice to iSPuP evaluation. Then, this chapter presents the method used to evaluate the research questions and the obtained results. SDP data from three distinct contexts were used and an interview with their respective process managers was carried out. It is detailed in next subsections.

7.2 Study Definition

The evaluation scope was defined based on GQM method (BASILI, 1994) as follows:

Analyze iSPuP approach and PROV-SwProcess provenance model to evaluate its feasibility

for the purpose of supporting data analysis and data-driven decision making

with respect to provide relevant information

under the point of view of process managers

in the context of software development process.

From the scope definition, the research questions are: *What are the analysis possibilities that can be carried out on the captured data?* and *How SDP analysis can help in process manager decision-making?*

[26] The third research question (***RQ3***. *What are the characteristics and limitations of the existing provenance approaches / models that deal with SDP provenance?*) was analyzed in Chapter 3.

7.3 Study Planning

- **Context selection**

Three distinct contexts were chosen for the approach evaluation.

The first process, called **SDP1** in this study, is used to manage change requests in a business management software. The company where this process was executed can be considered small, having among 10 to 49 employees, and is operating for more than ten years in the software development context.

The second process, called **SDP2**, deals with error handling and the implementation of new features in an Enterprise Resource Planning (ERP) system. It is from a medium-size company (having 59 employees) that acts in the software development context for more than ten years (about 24 years). A differential of this company is that all its employees work in home-office.

Finally, the third process, **SDP3**, deals with different issues as regards to developing and maintaining the company projects. This company is operating for more than ten years and can be considered as a large company, having more than 100 employees.

Despite the use of three different scenarios, it should be emphasized that the selected scenarios did not address the SDP as a whole. They deal with a subprocess of SDP, that deals with changes management / issue management.

- **Subjects characterization**

For each analyzed process, a subject was defined to evaluate the obtained results, using an interview (its script is in APPENDIX G). This subject selection considers only participants with a greater degree of knowledge about the process in the companies (managers with greater responsibility for the process):

SDP1: The selected subject is a male, who acted as a manager in SDP1 for two years. He holds a graduation degree in Information Systems.

SDP2: The selected subject is a male, who acted as a development manager for four years, and in the last eight years he is the company's development director. He was one of the responsible for creating SDP2 and directly monitors it since then. Considering these facts, he has a broad knowledge of the analyzed process and holds a master's degree in Computer Science.

SDP3: The selected subject is a male, who acted as a developer, a team leader and, lastly, as a process manager (for 2.5 years) in the SDP3 company, having a broad

knowledge of the analyzed process and provided data. He holds a PhD in Systems Engineering and Computer Science.

- **Computacional Support**

The case study was carried out on a PC configured with an Intel Core i5 CPU, 3 MB cache, 1.80 GHz processor, 6 GB RAM memory, shared video memory, 500 GB hard drive, and Microsoft Windows 8.1 Single Language, 64 bits.

7.4 Study Execution and Analysis

The study execution and analysis considering each of the three distinct contexts (SDP1, SDP2, and SPD3) is presented in the following subsections.

7.4.1 SDP1

- **Goals**

Analyze the proposed approach in supporting SDP execution data analysis and data-driven decision-making using real provenance data from 25 instances of a SDP.

- **Specific Scenario**

The analyzed data is from a SDP that manages change requests in a business management software. It should be emphasized that this is a specific software process to deal and control software changes and not a process to develop a completely new software. From the data requested to the company, they did not provide the procedures and resources used, and did not inform the names of the stakeholders involved in the execution of the activities. They only inform the names of the teams that performed them. Figure 7.1 shows the process flow model with its activities and roles. This model was used to capture process prospective provenance. From this process model, we obtained data about three specific activities: *Request the opening of Change*, *Solution Implementation*, and *Change RDM*[27] *to Complete*.

[27] RDM is the abbreviation for 'Request for Change', in Portuguese.

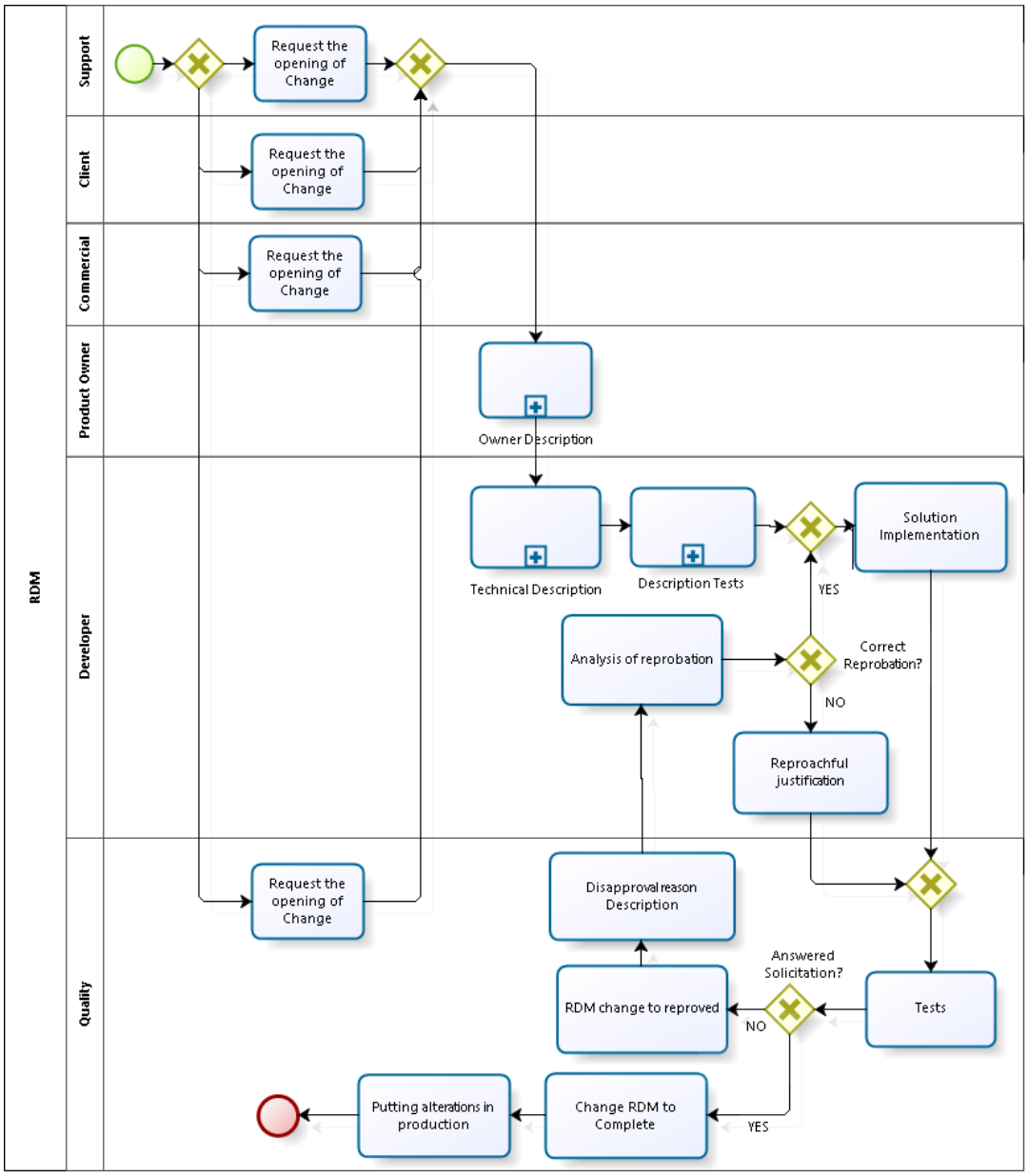


Figure 7.1: SDP 1 – Flow Model with Activities and Roles.

- **Execution**

The generated provenance graph with the SDP data from the 25 process instances is shown in Figure 7.2. The three stakeholders are represented by the orange pentagons, executed activities are the blue rectangles, and the artifacts correspond to the yellow ellipses. Considering that the amount of analyzed data is large (25 instances), several filters were applied into this graph visualization to facilitate its interpretation, besides the use of a tabular view provided by the tool[28].

[28] Hovering the mouse on each of the nodes of this graph, a tooltip is displayed with its details; besides that, there is the possibility of filtering this graph using node' type or name.

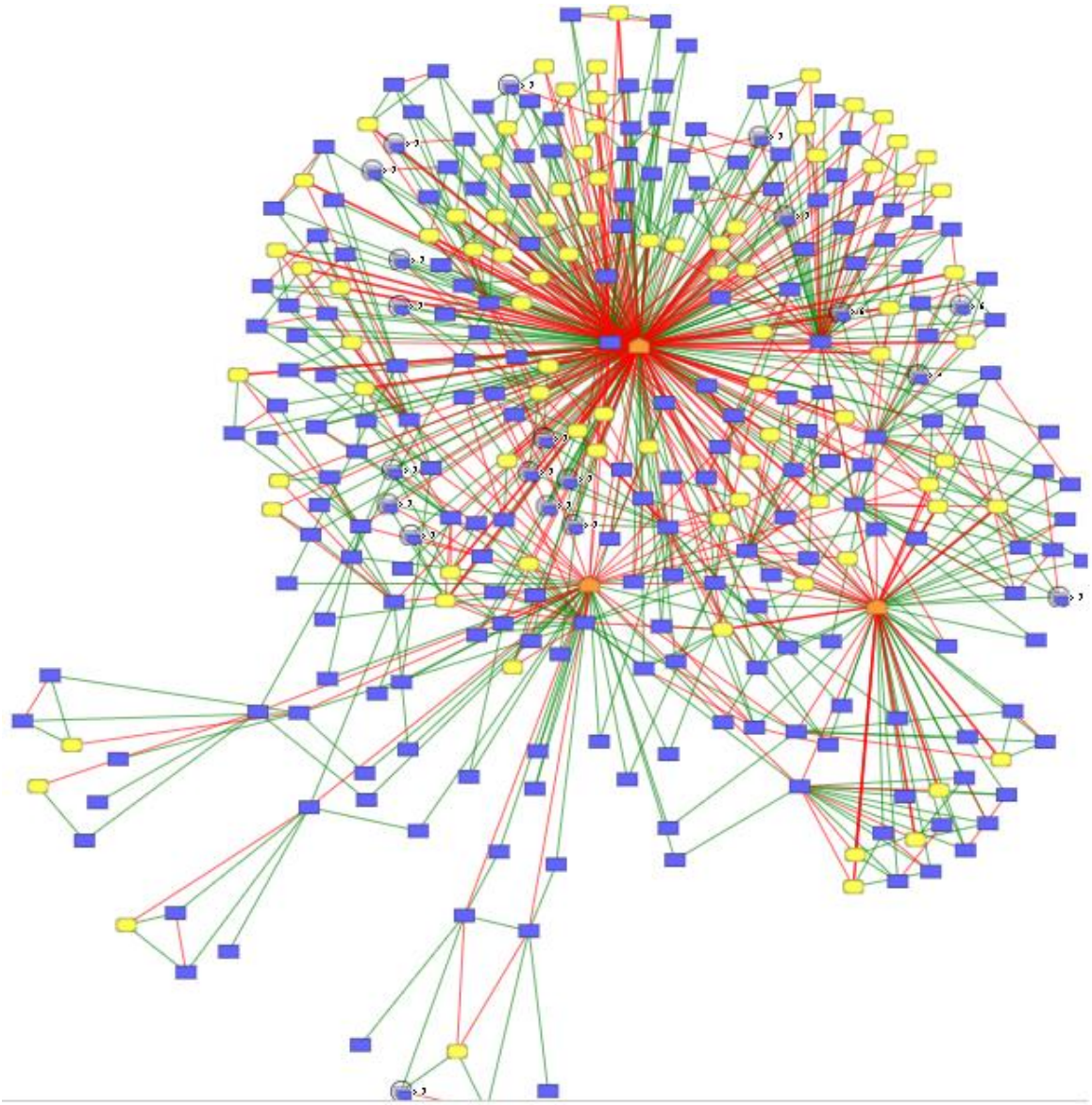


Figure 7.2: SDP1 – Twenty-Five Instances Overview.

A discussion about all the PROV-SwProcess CQs is presented in the following, using data from SDP1. For each CQ, four points were evaluated during the interview: (a) CQ analysis correctness, (b) check if the analyzes can assist in a previously defined decision-making, (c) check if the CQ can be answered using the company's current process management tool or dashboard, and (d) evaluate the relevance of answering the CQ to support in analysis and decision-making processes. All these four points were verified during the interview with the managers. Each of these points was discussed with the manager, after showing him the visualizations that supports in answering the CQs.

Goal 1: Process structure identification during execution and possibilities for process redesign

– **CQ1** *What are the process activities, artifacts, resources, procedures, stakeholders, and the relations among them during the process execution?*

Using iSPuP tool support, we can show all the process elements from the 25 instances and can also use some filters and *tooltips* that improve the process analysis. The macro visualization considering SDP is shown in Figure 7.2 and a tabular view using the same data (filtering by stakeholders) are in Figure 7.3. Using both visualizations we can see a discrepancy in the associated activities and manipulated artifacts among the three stakeholders (in orange pentagons) who participated in the process.

Name	Type	Degree	Created Artifacts	Modified Artifacts	Associated Activities
VB6	Team_Stakeholder	254	69	40	161
DotNet	Team_Stakeholder	57	13	11	33
Quality	Team_Stakeholder	50	0	0	25

Figure 7.3: SDP1 - Stakeholders X Activities – Tabular View.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is correct, because most part of the system is maintained by the *VB6 team* (about 95%) and only a small part of it has been developed using the .NET Framework[29] (this part is maintained by the *DotNet team*).

b) This analysis can *partially* assist in the proposed decision-making. In order to fully address the decision-making possibility, he would need that the tool exports those results to an excel format, where he could apply other filters and analyze these data and relationships more accurately.

c) He *cannot* answer CQ1 using his current process management tool (a proprietary tool developed in this same company).

d) Answering CQ1 is *very relevant* to support in analysis and decision-making processes.

[29] https://www.microsoft.com/net

— **CQ2** *Which procedures are used by the process during its execution?*

Using SDP1 provided data, CQ2 was not possible to be answered since no procedure was informed in the process execution data. However, we show to the process manager an analysis possibility in CQ2 using the toy example, and the following answers were obtained from him (we do not make the question to check if the analysis is correct, considering he does not know in details the process used in the toy - only a quick explanation of it was provided at the beginning of the interview):

a) -
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ2 using his current process management tool.
d) Answering CQ2 is *extremely relevant* to support in analysis and decision-making processes.

— **CQ3:** *Which activities had a high complexity (considering the number of associated stakeholders, artifacts, procedures and / or resources)?*

In order to answer CQ3, the activity's degree was used. Figures 7.4 and 7.5 are generated by the visualization tool to support CQ3. The tabular view was used, and we filter all the activities. After that, we ordered the obtained results by their degree - firstly descending (Fig. 7.4) and, after that, ascending (Fig. 7.5). The first three results obtained are shown in these figures. According to what is shown, it was not possible to perceive any great discrepancy between the levels of activities analyzed (minimum 1 and maximum 3), therefore, through this analysis, it was concluded that, according to this specific metric (activity grade), none of the activities performed during the 25 instances of the process can be considered more complex than the others.

id	Name	Type	Degree
8	Solution_Implementation_7213	Activity	3
9	Solution_Implementation_7212	Activity	3
10	Solution_Implementation_7211	Activity	3

Figure 7.4: SDP1 – Activities Degree – Part 1.

id	Name	Type	Degree
52	Opening_the_request_for_change_7202	Activity	1
134	Opening_the_request_for_change_7159	Activity	1
149	Opening_the_request_for_change_7274	Activity	1

Figure 7.5: SDP1 – Activities Degree – Part 2.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*.
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ3 using his current process management tool.
d) Answering CQ3 is *extremely relevant* to support in analysis and decision-making processes.

– **CQ4:** *Which activities had a high dependency (on other activities)?*

Activities dependency on other activities is provided by PROV-SwProcess Model using the relation *WasInformedBy* (implying that there has been the exchange of some artifact by two activities, one activity using (or changing) some artifact generated by the other activity) and is useful only when just one instance is analyzed. When checking SDP1 instances separately, no dependency between activities was found. This fact can be explained because in SDP1 data, only information related to artifacts handled during the *Solution Implementation* activity was provided. For the other activities, no manipulated artifacts were mentioned. We discussed this fact with the manager, and the following points should be considered:

a) This fact is *correct*. The manager mentioned that the company should store data about which artifacts were analyzed and tested - in fact - by the quality team during their test activity, and it would be interesting to be able to track the associations between the development and test activities.
b) It *can* assist in the proposed decision-making.
c) He *cannot* answer CQ4 using his current process management tool.
d) Answering CQ4 is *extremely relevant* to support in analysis and decision-making processes.

Goal 2: Understanding stakeholder's involvement in process execution

– CQ5: *What is the activities distribution among stakeholders?*

As already mentioned in CQ1, *VB6* performed much more activities than *DotNet* and *Quality* (Figure 7.2 and 7.6). Besides that, as shown in Figure 7.7, *Quality* is only associated with 25 identical activities (*Set_Change_Request_Completed* - the name of all 25 activities is displayed when hovering the mouse on the activity grouping), when considering their associations with other process elements (and, for this reason, they were grouped).

Name	Type	Degree	Created Artifacts	Modified Artifacts	Associated Activities
VB6	Team_Stakeholder	254	69	40	161
DotNet	Team_Stakeholder	57	13	11	33
Quality	Team_Stakeholder	50	0	0	25

Figure 7.6: SDP1 - Stakeholders X Activities – Tabular View.

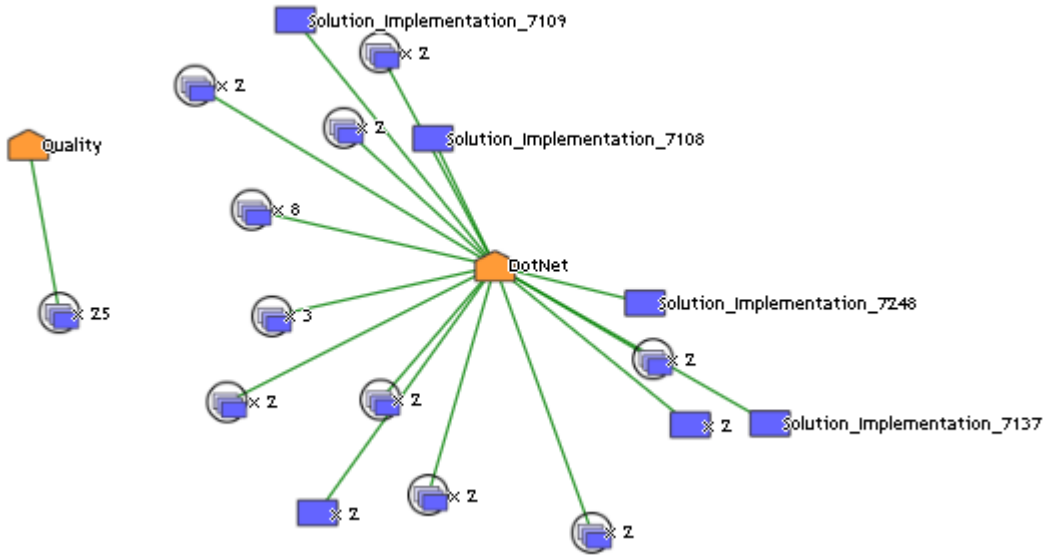


Figure 7.7: SDP1 - Stakeholders X Activities – Quality and DotNet activities.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*, considering the company has two development teams (*VB6* and *DotNet*) and *VB6* has much more developers than DotNet - because the system is 95% implemented in VB6. Besides that, quality team is always responsible for verifying all the tasks performed

by both teams and, therefore, is associated with 25 activities (the same number of process analyzed instances).

b) This analysis *can* assist in the proposed decision-making.

c) He *can* answer CQ5 using his current process management tool.

d) Answering CQ5 is *not very relevant* to support in analysis and decision-making processes. When he gave this response, the manager mentioned that only with team stakeholders, this analysis was not very relevant to support him in decision-making. It does not bring any novelty. Ideally, it would be possible to analyze all the participants (persons) of the process.

– **CQ6:** *Which artifacts are known by a stakeholder, considering that in some process execution he/she created or modified such artifact?*

Figures 7.8 and 7.9 are examples of visualizations generated by the visualization tool to support answering CQ6. According to Figure 7.9, we can see, for example, that only *DotNet* stakeholder manipulated some artifacts like *Forca_de_Venda_-_PDA-Pedidos.prj* and *Gerenciador-WebNovoCodigo.vb* and, therefore, we may consider that *DotNet* has some knowledge about them. On the other hand, artifacts like *Genciador-clsFuncao* and *Gerenciador-Dados.vb* were manipulated both by *DotNet* and *VB6*, for example. Another analysis that should be considered is that *Quality* stakeholder did not manipulate any artifact in the analyzed SDP instances. Based on this, in a future instantiation of the analyzed process, if a certain task is associated with a specific artifact (*Gerenciador-Dados.vb* for example), the process manager can allocate this task to *DotNet* or *VB6*, considering that both know this artifact. On the other hand, if an activity needs to use the artifact *Gerenciador-WebNovoCodigo.vb,* he/she can allocate this task to *DotNet*, that previously manipulated it, which may possibly contribute to the task being carried out more quickly.

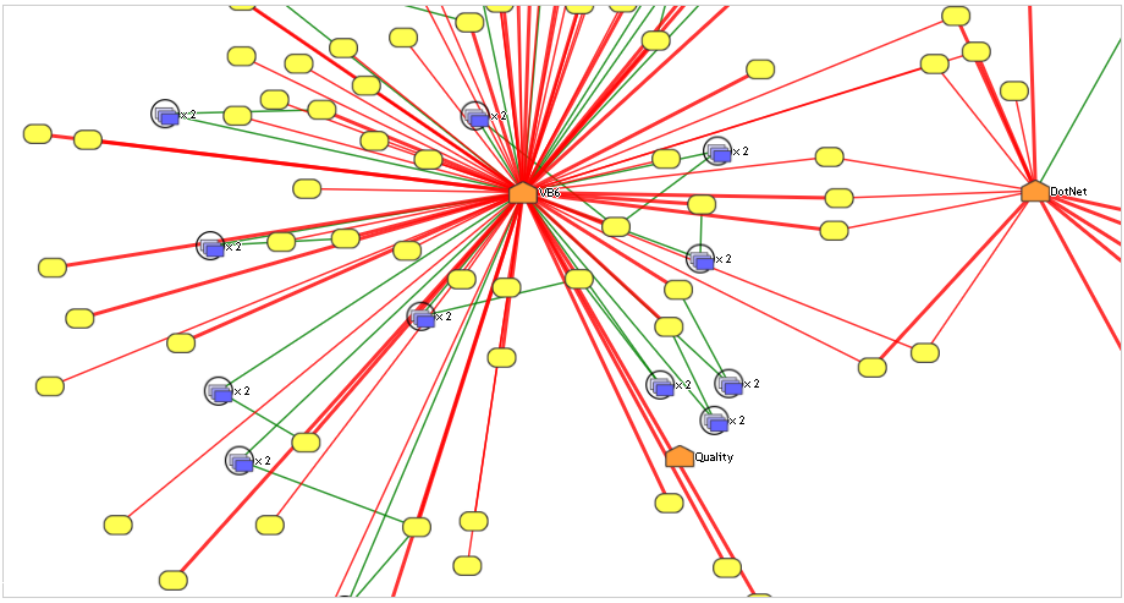


Figure 7.8: SDP1 – All Stakeholders X Artifacts.

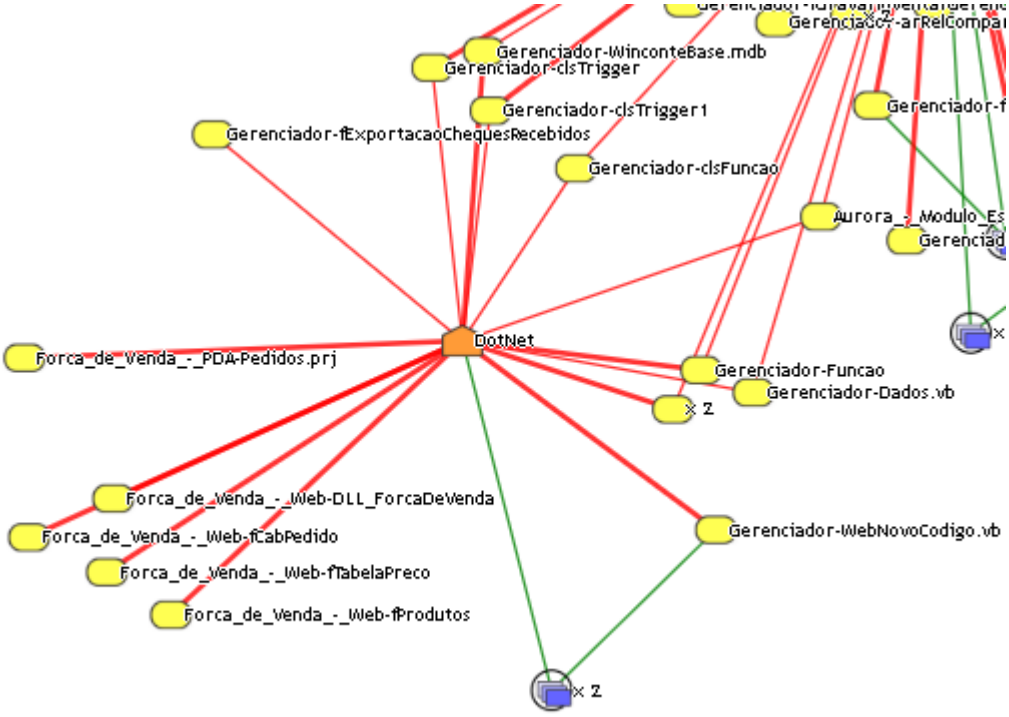


Figure 7.9: SDP1 - DotNet Associated Artifacts

When this analysis was presented to the manager, the following answers were obtained:

a) The presented analysis is *correct*.

b) This analysis *can* assist in the proposed decision-making.

c) He *can partially* answer CQ6 using his current process management tool.

d) Answering CQ6 is *very relevant* to support in analysis and decision-making processes.

— **CQ7:** *Which stakeholders are out of the average of created and/or modified artifacts?*

Figure 7.10 is generated by the visualization tool to support in answering CQ7. *VB6* created and modified much more artifacts than other stakeholders, and this same stakeholder created 69 artifacts while modified only 40. *DotNet* created 13 and modified 11 artifacts, while quality did not modify or create any artifact.

id	Name	Type	Degree	Created Artifacts	Modified Artifacts
343	VB6	Team_Stakeholder	254	69	40
192	DotNet	Team_Stakeholder	57	13	11
227	Quality	Team_Stakeholder	50	0	0

Figure 7.10: SDP1 - Stakeholders X Created and Artifacts – Tabular View.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*. He mentioned that although the quality team manipulates some artifacts to test them, the information of which artifacts have been used is not stored yet.
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ7 using his current process management tool.
d) Answering CQ7 is *somewhat relevant* to support in analysis and decision-making processes, because the creation or modification of artifacts is directly related to the requested demands (creation of new functionalities of correction of errors in the system) and this should be addressed in this analysis.

— **CQ8:** *What are the relationships among stakeholders?*

Using SDP1 provided data, CQ8 was not possible to be answered since no relation among stakeholders' roles was informed in the process execution data and this

information was not inferred by PROV-SwProcess. However, we showed to the process manager an analysis possibility in CQ8 using the toy example, and the following answers were obtained from him:

a) -
b) This analysis *can* assist in the proposed decision-making.
c) He *can partially* answer CQ8, considering that "in medium-size companies it is easy to detect and the relationships among stakeholders do not change, i.e., it does not have any variations during process execution instances".
d) Answering CQ8 is *somewhat relevant* to support in analysis and decision-making processes, because he believes that it could be very or extremely relevant only in large companies.

— **CQ9:** *Which roles does each stakeholder assume?*

CQ9 was not possible to be answered using SDP1 provided data, because the role performed by a stakeholder when he/she was associated with an activity, was not provided. Besides that, the names of the stakeholders involved in the execution of the activities were not informed (they only inform the names of the teams that performed them – *VB6*, *DotNet*, *Quality*). Then, we showed to the process manager an analysis possibility in CQ9 using the toy example, and the following answers were obtained from him:

a) -
b) This analysis *can* assist in the proposed decision-making.
c) He *can partially* answer CQ8, considering what was already mentioned in the previous question ("in medium-size companies it is easy to detect and it does not have any variations during process execution instances").
d) However, the knowledge about the roles that can be played by each stakeholder is *extremely relevant* to support in analysis and decision-making processes.

Goal 3: Tracking derivations and revisions among artifacts or procedures

— **CQ10:** *Which artifacts are derivations from others?* and

— **CQ11:** *Which artifacts or procedures are revisions from others?*

Figure 7.11 is generated by the visualization tool to support the achievement of GOAL 3 (CQ10 and CQ11). Considering this visualization, no derivation between artifacts was found (no association was inferred among the artifacts manipulated by the 25 instances of this process). This fact occurs because artifacts used in the activities were not informed, and, then we cannot infer them. Process data only has the artifacts created and changed by the activities. Then, the possibilities of decision-making to achieve this goal cannot be applied to this process.

When this analysis was presented to the manager, the following answers were obtained (both for CQ10 and CQ11):

a) This analysis is *correct*, considering only the analyzed group of data.

b) This analysis *can* assist in the proposed decision-making.

c) He *cannot* answer CQ10 and CQ11 using his current process management tool.

d) Answering CQ10 and CQ11 is *extremely relevant* to support in analysis and decision-making processes.

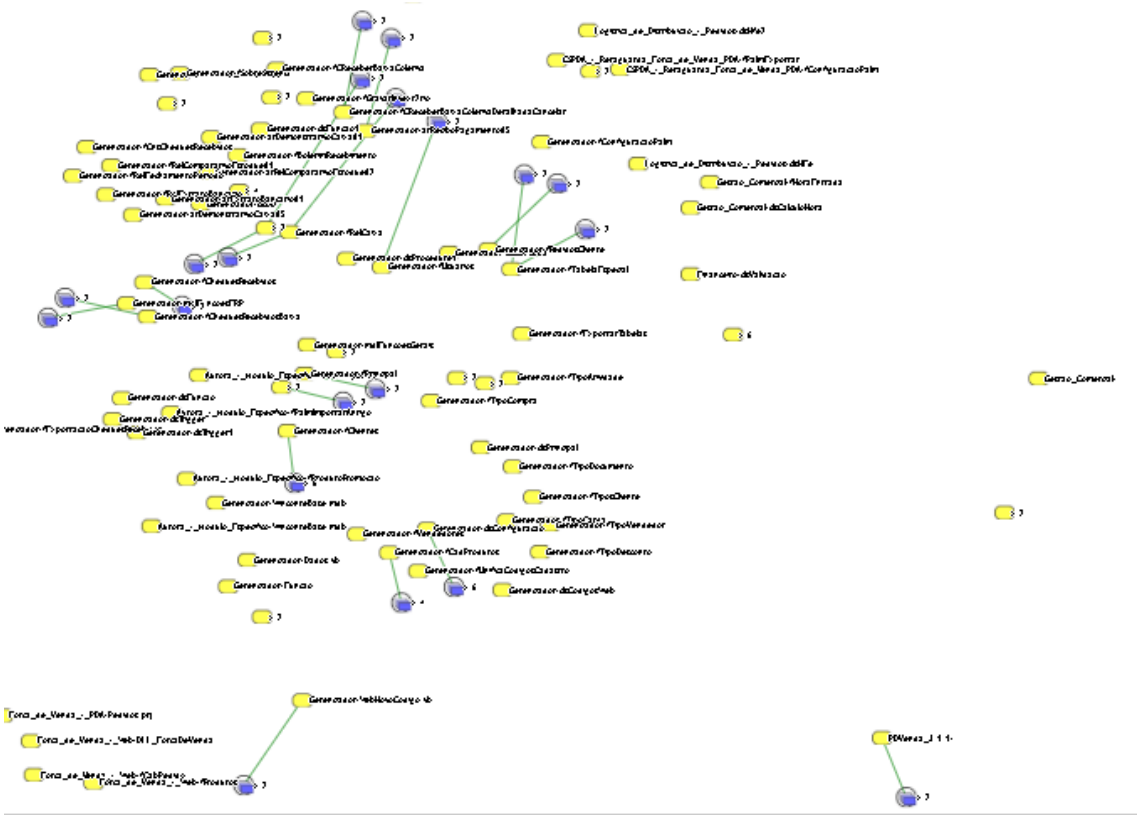


Figure 7.11: SDP1 – Artifacts Derivation.

- **Analysis Summary**

  Considering SDP1, in summary, the following results were obtained:

- Seven of the eleven CQs proposed by PROV-SwProcess model could be answered with the dataset provided by the company. The four that could not be answered (CQ2, CQ8, CQ10, CQ11) are due to the fact that insufficient data were provided to make them possible (the procedures used during the execution of the process, the relationship between the roles, and some information that enables capturing the derivations and revisions between the artifacts and/or procedures were not informed). The absence of such data was recurrent in the next two analyzed processes (SDP2, SDP3).
- When verifying with the manager the **correctness** of each of the seven CQs that were possible to be answered, 100% of them were evaluated as correct.
- When considering if the analyzes **can assist on decision-making** (according to each CQ), the manager mentioned *yes* for 10 CQ and *partially* only for once (CQ1).
- By questioning if the manager could **answer the CQs** using his current management tools and dashboards, he said *no* for seven CQs, *partially* for three and *yes* just for one of them.
- The manager also evaluates the relevance of each CQ to support in process analysis and decision making. He considered 6 of them *Extremely relevant*, 2 are *Very relevant*, other 2 are *Somewhat relevant*, and just one was considered *Not very relevant*. None CQ was evaluated as *Irrelevant*.
- Considering the final group of questions of the interview script (APPENDIX G), the manager initially said *yes* when asked if the presented CQs were adequate and sufficient to achieve the proposed goals; however, after thinking a little more about this question, he suggested to change the answer to *partially* because, although CQs are adequate, he believes that GOAL 2 should take into account, at some point, the type of tasks performed (maintenance or new resource implementation) to better understand stakeholder's involvement in process execution. The researcher mentioned to him that a simple filter by type of activity in the visualization tool would solve this question and he agreed.
- As new questions to be analyzed the manager suggested the creation of two new possibilities: (1) Analyze the 'failure rate' of the development activities performed (if data related to the test activities were provided); (2) Implement

filters that allow to display the relationship between each process instance, the versions associated with it, and the ‘affected’ clients on each process instance.

- The manager did not have comments on the interview.
- As regards to the approach as a whole, he mentioned that he found it quite interesting to analyze what is happening in the process, however, the tool needs to offer more filters and reporting possibilities (spreadsheets export) that allow the manipulation of this data, besides the shape in graph and just a single table.

7.4.2 SDP2

- **Goals**

Analyze the proposed approach in supporting SDP execution data analysis and data-driven decision-making using provenance data from 10 instances of a real-world SDP.

- **Specific Scenario**

The analyzed data is from a process that deals with error handling and implementation of new features and in an ERP Project. It is performed by six different roles (Client, Test Team, Support, Support Manager, Development Manager, and Programmer) and is from a company expert in creation / maintenance of accounting systems and is in the market for more than 25 years. From the data requested to the company, they did not provide the procedures and resources used. Stakeholder’s names have been hidden to preserve their privacy. Figure 7.12 shows the used process flow model with its activities and roles. This model was used to capture process prospective provenance. From this process model, we only obtained data about five specific activities: *System Error Report, New Feature Request, Case Registration, Case Resolution,* and *Close the Case.*

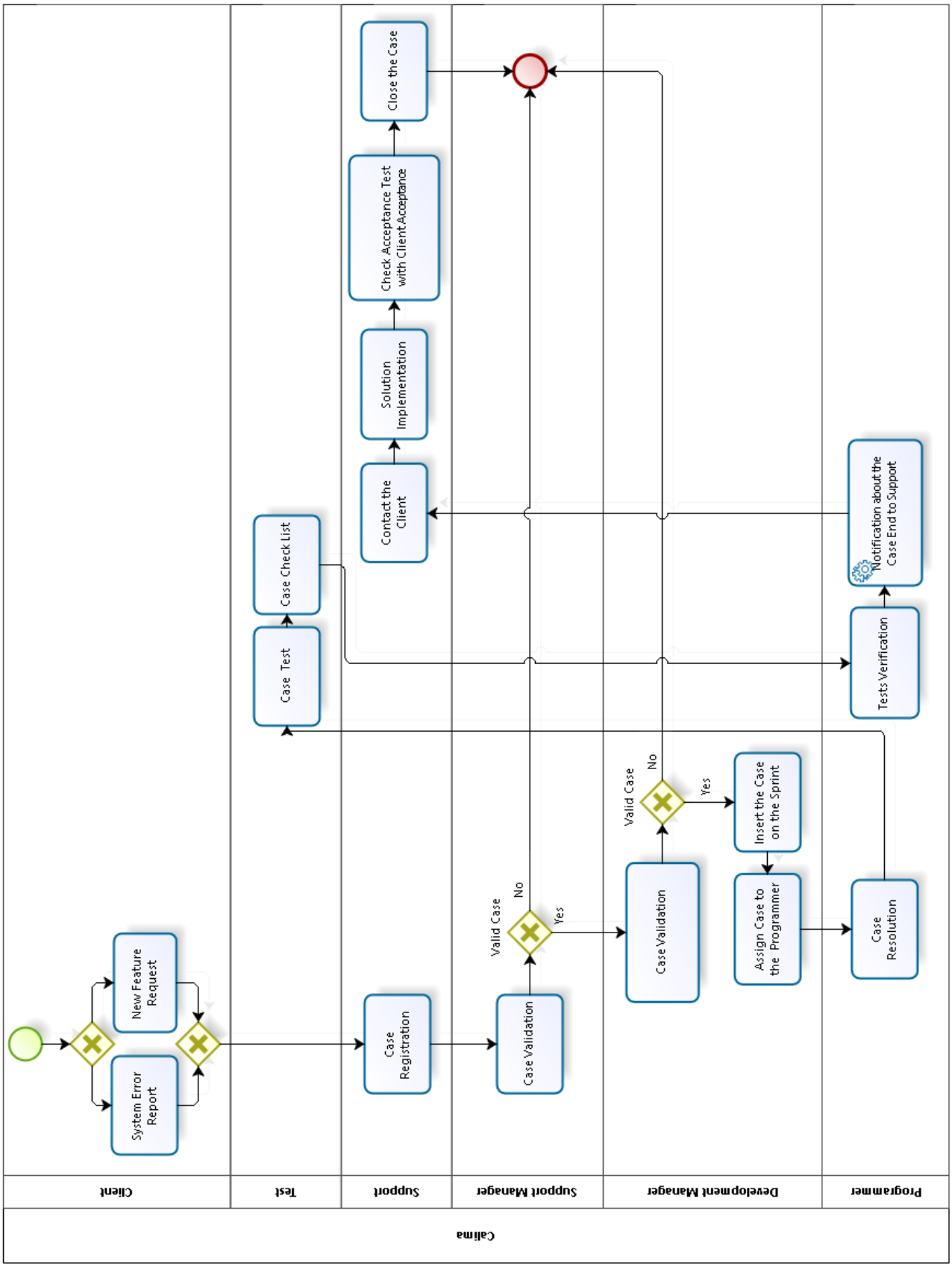


Figure 7.12: SDP2 – Flow Model with Activities and Roles.

- **Execution**

The generated provenance graph with the SDP data from the 10 process instances is shown in Figure 7.13. The reported stakeholders are represented by the orange pentagons, executed activities are the blue rectangles, and the artifacts and roles correspond to the yellow ellipses. After the generation of the visualization presented in this figure, filters were applied into this graph to facilitate its interpretation in order to answer CQs, besides the use of a tabular view.

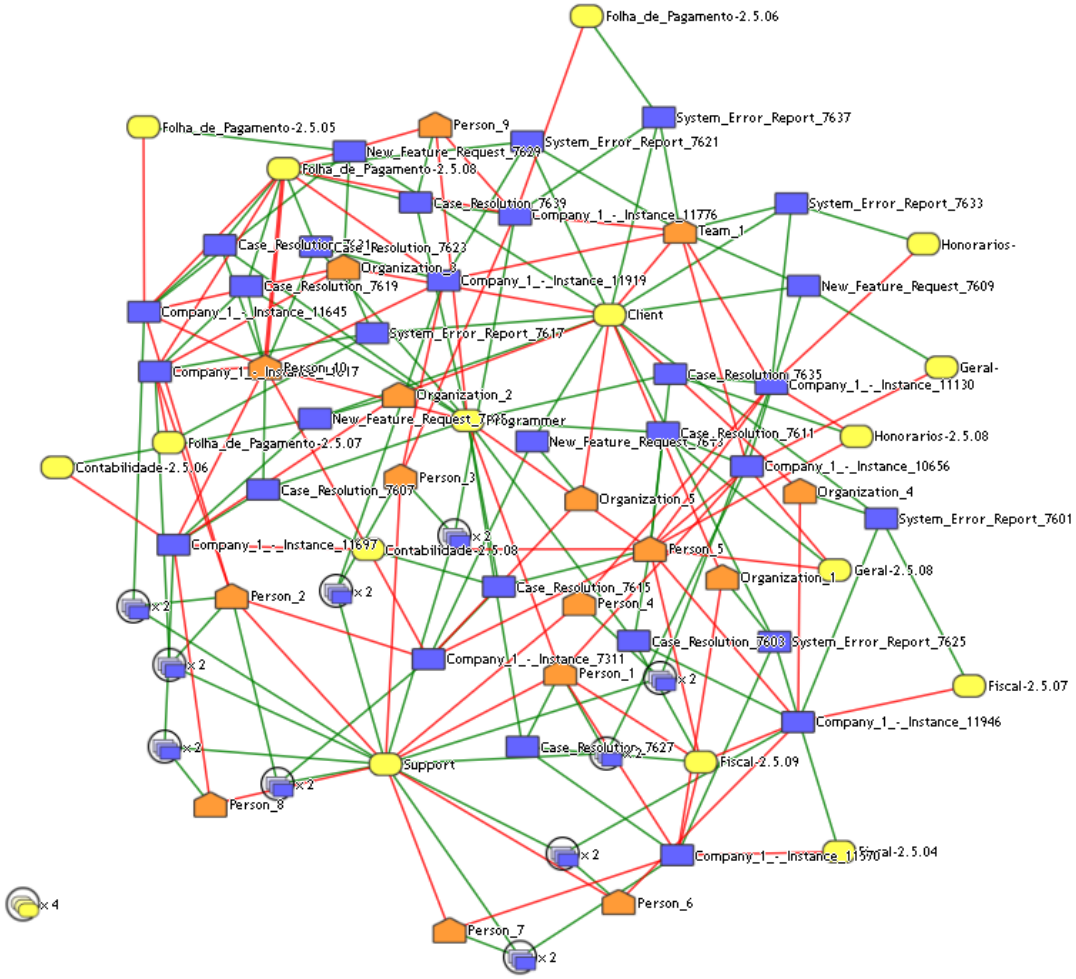


Figure 7.13: SDP2 – Ten Instances Overview.

A detailed discussion about all the PROV-SwProcess Competency Questions is presented in APPENDIX H, using data from SDP2 and including the manager's opinion on each of them (the same procedure reported for SDP1 was followed, changing only the scenario and the subject).

Considering SDP2, in summary, the following results were obtained:

- When verifying with the manager the **correctness** of each of the seven CQs that

were possible to be answered, 100% of them were evaluated as correct.

- When considering if the analyzes **can assist on decision-making** (according to each CQ), the manager mentioned *yes* for 10 CQ and *partially* only for once (CQ8).
- By questioning if the manager could **answer the CQs** using his current management tools and dashboards, he said *no* for five CQs, *partially* for two and *yes* for four of them. The company responsible for SDP2 was most able to answer the CQs using their own management tools.
- Considering the relevance of each CQ to support in process analysis and decision making, the manager marked 4 of them *Extremely relevant*, 4 are *Very relevant*, 3 are *Somewhat relevant*, and none CQ was evaluated as *Not very relevant* or *Irrelevant*.
- Considering the final group of questions of the interview script (APPENDIX G), the manager said *partially* when asked if the presented CQs were adequate and sufficient to achieve the proposed goals, because he could not see GOAL 3 (*Track derivations and revisions among artifacts and procedures*) being achieved with the data provided by his company.
- As new questions to be analyzed, the manager suggested better exploring other possibilities of stakeholder relationships (not just acted on behalf of) and consider activities' time spent and their planned complexity.
- The manager did not have comments on the interview.
- With regard to the approach as a whole, the manager said "the relations and inferences between the process elements shown by the approach are quite interesting, however, we should analyze whether these represent 'outliers' or if they are actually occurring always, indicating some 'problem' in the process. The approach would be very useful for data-based consultancies".

7.4.3 SDP3

- **Goals**

Analyze the proposed approach in supporting SDP execution data analysis and data-driven decision-making using provenance data from 133 instances of a real-world SDP.

- **Specific Scenario:**

The analyzed data is from a process that deals with error handling and implementation of new functionalities in several projects. For the analyzes presented in the following, a specific project was chosen, which had 133 instances of the process shown in Figure 7.14. It is performed by three different roles (Reporter, Manager, and Developer) and has three main activities: *Issue Registration*, *Issue Attribution*, and *Issue Resolution*. From the data requested from the company, they did not provide the procedures and resources used. Stakeholder's names have been masked to preserve their privacy.

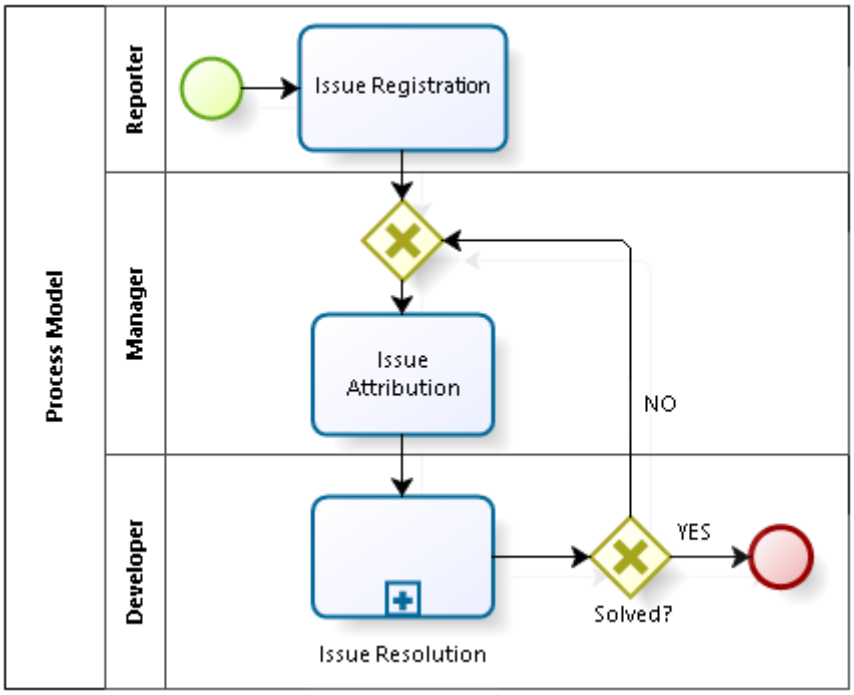


Figure 7.14: SDP3 – Flow Model with Activities and Roles.

- **Execution**

The generated provenance graph with the SDP data from the 133 process instances is shown in Figure 7.15. The stakeholders reported are represented by the

orange pentagons, executed activities are the blue rectangles, and the artifacts and roles correspond to the yellow ellipses. After the generation of the visualization presented in this figure, filters were applied into this graph to facilitate its interpretation in order to answer CQs, besides the use of a tabular view.

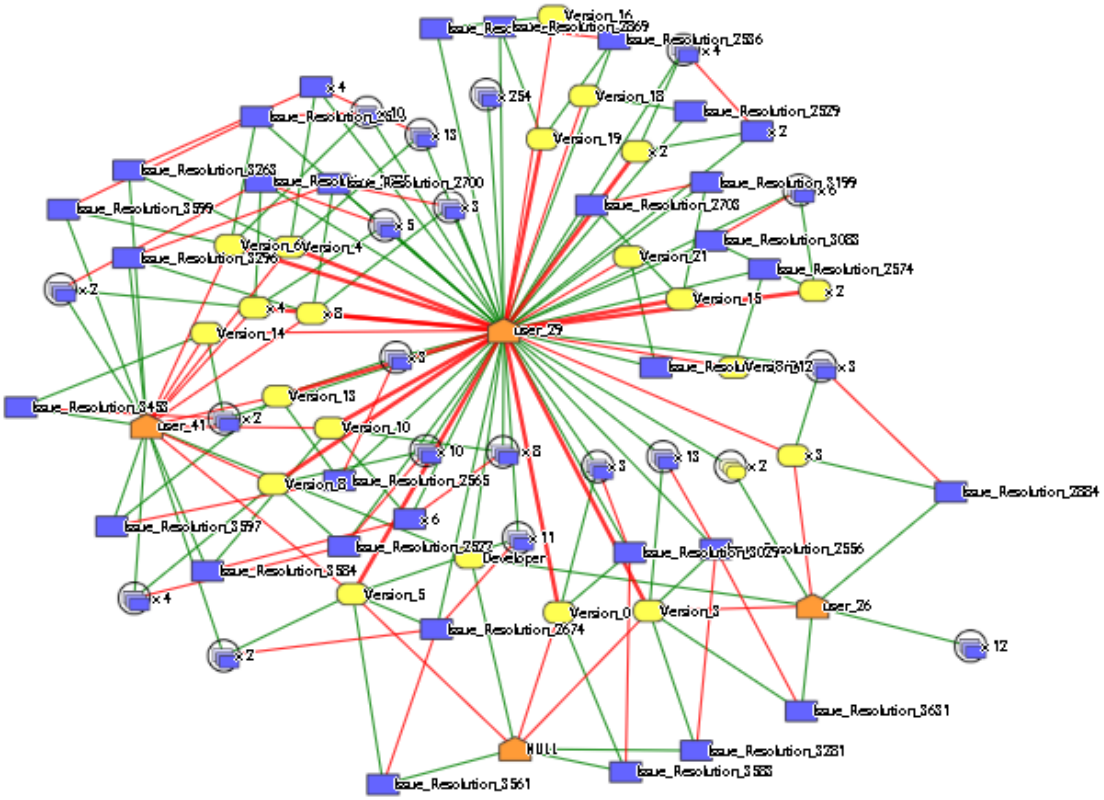


Figure 7.15: SDP3 – One Hundred and Thirty-Three Instances Overview.

A discussion about all the PROV-SwProcess Competency Questions is presented in APPENDIX I, using data from SDP3 and including the manager's opinion on each of them (the same procedure reported for SDP1 was followed, changing only the scenario and the subject). Considering SDP3, in summary, the following results were obtained:

- When verifying with the manager the **correctness** of each of the seven CQs that were possible to be answered, 6 of them were evaluated as correct and just one was evaluated as partially correct (CQ3), because he believes that only the degree of the activity cannot be determinant to evaluate its complexity.
- When considering if the analyzes **can assist on decision-making** (according to each CQ), the manager mentioned *yes* for 9 CQ and *partially* for two (CQ5, CQ6).
- By questioning if the manager could **answer the CQs** using his current

management tools and dashboards, he said *no* for all the CQs.

- When evaluating the relevance of each CQ to support in process analysis and decision making, the manager considered 7 of them *Extremely relevant*, 3 are *Very relevant* and just one was considered *Not Very Relevant*. None CQ was evaluated as *Somewhat relevant* or *Irrelevant*.
- Considering the final group of questions of the interview script (APPENDIX G), the manager said *yes* when asked if the presented CQs were adequate and sufficient to achieve the proposed goals.
- He did not suggest other questions to assist in SDP analysis and decision making.
- The manager did not have comments on the interview.
- As regards to the approach as a whole, he pointed that it would be interesting include time in the analysis (using a slider to be aware of the changes in the process throughout the instances execution) and better exploit filter replication between the two proposed views – graph and tabular (the same filter, when applied, should serve both views).

7.5 Results Discussion

This section is a summary of the results obtained when considering the four points: (a) analysis correctness, (b) check if the analyzes can assist in a previous defined decision-making, (c) check if the CQ can be answered using the company's current process management tool or dashboard, and (d) evaluate the relevance of answering the CQ to support in analysis and decision-making processes. All these four points were verified during the interview with the managers, for each of the eleven CQ.

a) Evaluate the correctness of the performed analyzes using SDP data: Figure 7.16 shows that just one analysis in both processes was considered incorrect. All other analyses were considered correct. This result can be considered as evidence that the use of iSPuP approach and PROV-SwProcess model, when dealing with real process data, results in correct analyses. It should be noted that the analyses were carried out by a person who did not participate or manage the analyzed process (i.e., with iSPuP tool support, it is possible to perform the correct analysis without the need for in-depth knowledge of the process).

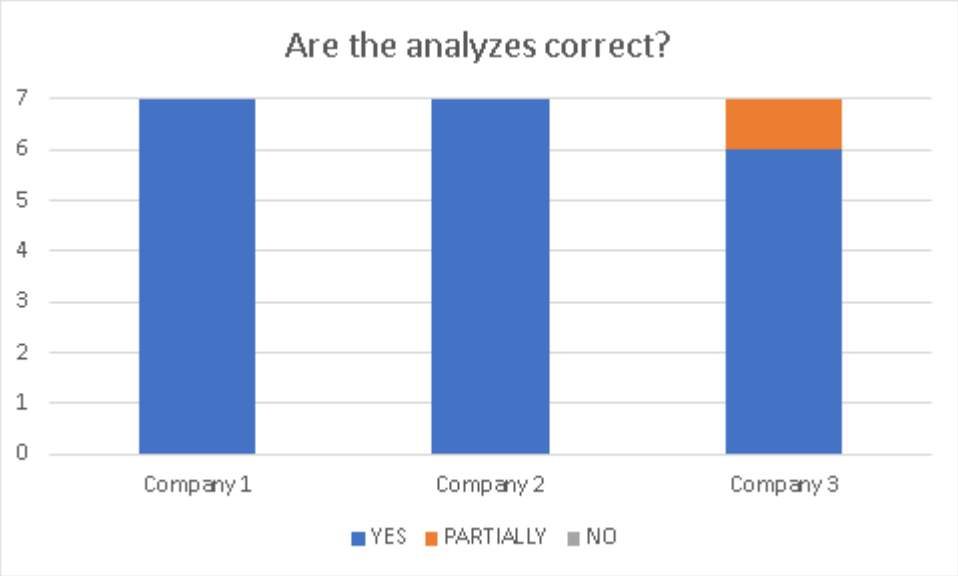


Figure 7.16: Results about the correctness of the analyses.

(b) Evaluate if the performed analyses can assist in the proposed decision making: When considering companies 1 and 2, Figure 7.17 shows that just one analysis (of the eleven that were performed) can *partially* assist in the proposed decision-making. In company 3, the manager states that two of them can *partially* assist in decision-making. It should be emphasized that for none of the CQs, the managers pointed that the analyses *could not* assist in the proposed decision-making. Regarding the questions in which the answer was *partially*, there was no consensus among the managers: Company 1 - **CQ1**, Company 2 - **CQ8**, and Company 3 - **CQ5** and **CQ6**. However, the managers from companies 2 and 3 cited the *lack of specific data* (e.g., activities duration, the level of stakeholder's relationship, and the level of artifacts knowledge) as a reason for not allowing the proposed decision making to occur completely. The manager of Company 1 states the need to export these data to a spreadsheet format, to allow him to better manipulate them.

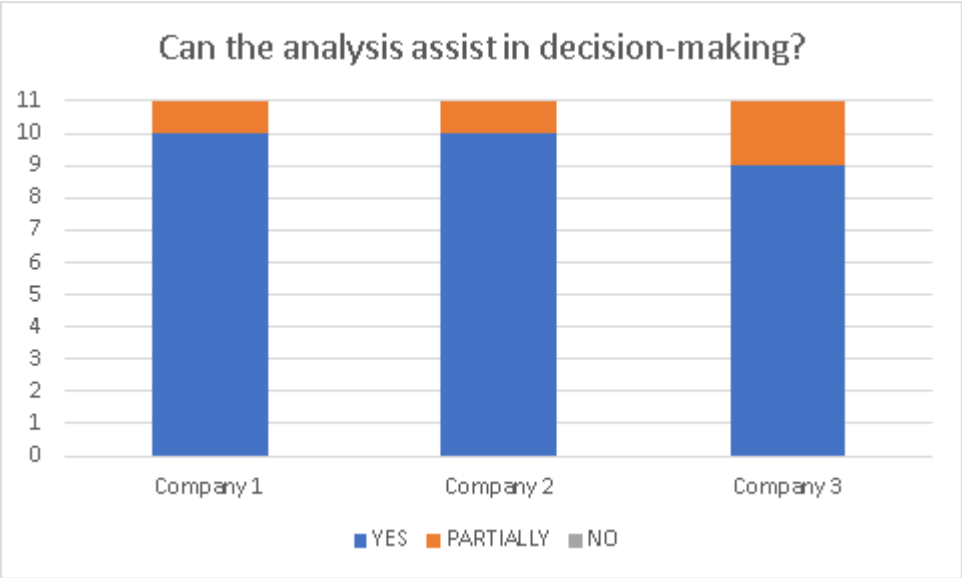


Figure 7.17: Results when checking if the performed analyses can assist in the proposed decision making.

c) Verify if the CQ can be answered using the company's current process management tool or dashboard: In Company 1, 63.3% of the CQ could not be answered using the company's current process management tool or dashboard. In Company 2, this rate is 45.4% and in Company 3 it is 100%, as shown in Figure 7.18.

A possibility raised about the Company 2 ability in obtaining more answers to the CQs could be because this is a company in which all the employees work in home-office, which requires a greater control and monitoring over the process activities, considering that there is no possibility of analyzing these through some personal contact.

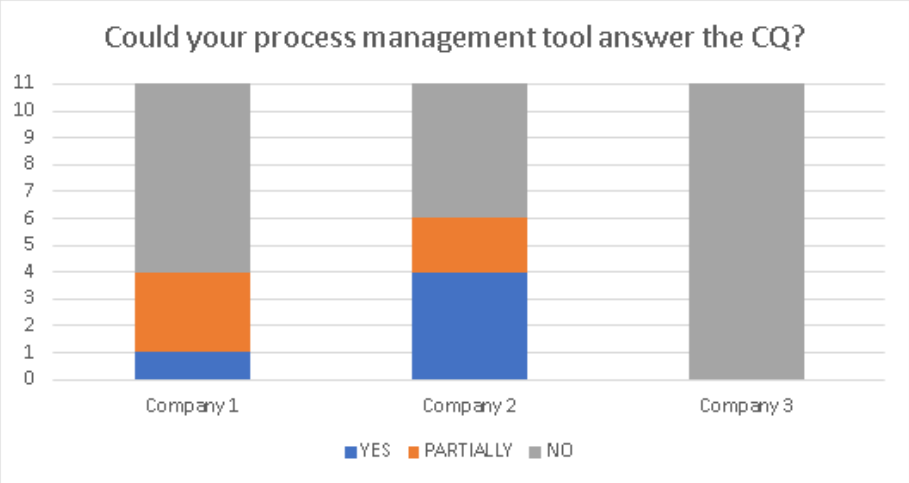


Figure 7.18: Results when checking if the process manager can answer the CQs using his current process management tool or dashboard.

d) Evaluate the relevance of answering the CQ to support in analysis and decision-making processes: As shown in Figures 7.19 and 7.20, answering the proposed questions is extremely relevant to aid decision making in 52% of the cases. For none of the CQ, answering it would be *irrelevant* to the proposed decision-making.

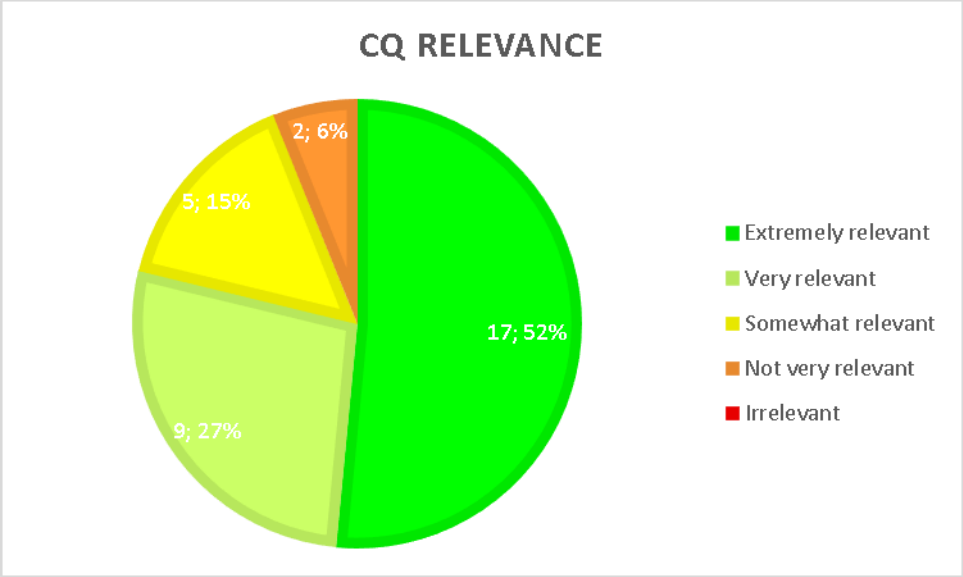


Figure 7.19: CQ Relevance.

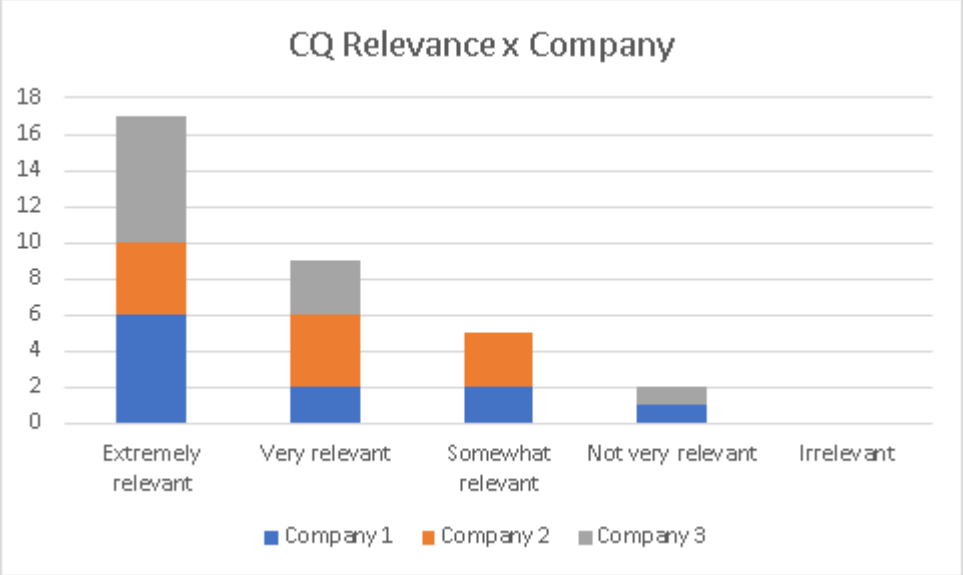


Figure 7.20: CQ Relevance X Company.

Considering the above results and the two research questions presented in Chapter 1 (***RQ4***. *What are the analysis possibilities that can be carried out on the captured data?* and ***RQ5***. *How SDP analysis can help in process manager decision-making?*), the following considerations should be cited:

- This chapter has shown that the analysis possibilities presented in Chapter 4 on the captured data can be performed in real scenarios, except in cases where insufficient data have been provided for them;
- Decision-making possibilities on each CQ have been carefully presented and evaluated by managers, as presented above.

7.6 Threats to Validity

Considering the subjects selection of the case study, as threats to validity, we can mention the reduced number of them (just one for each SDP). This threat was anticipated, and we tried to soften it by choosing the subjects with a greater degree of knowledge about the process in the companies (managers with greater responsibility for the process), since it would not be possible to carry out the evaluation with all the managers involved in the process.

Despite the use of three different scenarios in which the approach was applied, the results cannot be generalized to all software processes. It should be emphasized that the selected scenarios did not address the software development process as a whole, they deal with software changes management / issue management. Based on this, it is necessary to prepare and conduct additional experimental studies to extend the validity of this thesis' hypothesis (*The use of provenance models and techniques for capturing and analyzing software process provenance data can improve and assist process managers in the SDP analysis and support data-driven decision-making*). However, although the results cannot be generalized, it is possible to identify situations in which similar results can be achieved.

Further studies could collect additional evidence that was not observed in our regular case study. Besides that, additional experimental studies could also reveal certain aspects that were not considered such as non-functional requirements, e.g., performance and scalability, among others.

7.7 Final Remarks

This chapter presents the conducted evaluation to check iSPuP's ability in supporting SDP analysis and data-driven decision-making in real SDP contexts.

Considering this thesis hypothesis (*The use of provenance models and techniques for capturing and analyzing software process provenance data can improve*

and assist process managers in the SDP analysis and support data-driven decision-making), a case study with data from three different processes was carried out, showing that the use of the iSPuP approach, with PROV-SwProcess provenance model, is capable of assisting in making previously established decisions, and most of them would not be possible with the systems and tools currently adopted by the companies.

CHAPTER 8 – CONCLUSION

This chapter presents the thesis contributions and results, including open questions suggested as future works.

8.1 Epilogue

Companies have been increasing the amount of data they collect from their systems and processes, considering the dropping cost of memory and storage technologies in the last years. Traceability and provenance are promising approaches when considering the emergence of technologies such as Big Data, Cloud Computing, E-Science, and the increasing complexity of systems and processes. However, as we present in the literature review (Chapter 3), it is still rare in the literature mature proposals addressing the use of provenance in SDP. Besides that, no proposal covers all the specificities of software processes, including its main elements (activities, artifacts, stakeholders, resources and procedures) and the provenance relationships between them. To this end, PROV-SwProcess (a provenance model for software processes including its main elements, relations, inference rules and competency questions) was developed and evaluated by experts in process and provenance area (Chapter 4). This chapter also details a series of competency questions (CQs) that PROV-SwProcess is able to answer (using an ontology).

In order to support PROV-SwProcess model instantiation (allowing to structure all the process execution data according to PROV-SwProcess model), strategical information discovering (through inferencing mechanisms) and data visualization (allowing process data analysis and managers' data-driven decision-making), iSPuP approach was defined and its tool support was developed (Chapter 5).

Three provenance and process experts evaluated PROV-SwProcess model (Chapter 6), allowing to correct some model defects. Their suggestions and corrections were incorporated in the model current version. In the last round of this evaluation, the expert pointed out 32 correct points and 6 defects (3 incorrect facts, 1 inconsistence and 2 omissions).

The case study, presented in Chapter 7, showed that iSPuP approach have the potential to improve and assist process managers in the SDP analysis and support data-driven decision-making, using PROV-SwProcess Provenance Model. Most of the

proposed decisions would not be supported / possible using the systems and tools currently adopted by the companies.

In summary, this thesis (i) pointed out drawbacks and gaps in current provenance models to deal with SDP domain; (ii) proposed and evaluated a provenance model for SDP, including an operational ontology with inference rules and competency queries; (iii) presented iSPuP approach, its main phases and architecture elements, enabling PROV-SwProcess model instantiation, new information inferencing and data visualization; and (iv) provided initial evidence on the use of the approach and the provenance model.

8.2 Contributions and Results

The research and work described in this thesis has the following contributions:

- *A Quasi*-Systematic Literature Review of Provenance in the Context of Software Development Processes;
- A provenance model (with an operational ontology) to accommodate SPD provenance specificities, including its main elements, relations, inference rules and competency questions;
- A set of competence questions that can be answered with the proposed model and the respective decision-making possibilities that can be performed in answering these questions;
- An approach (called iSPuP) with tool support to instantiate PROV-SwProcess model with process provenance data, new information inferencing, and data visualization (allowing data analysis and decision-making); and
- The iSPuP evaluation, using three real scenarios with software process data execution.

8.2.1 Research achievements

The conduction of this research allowed the following research achievements:

- DALPRA, H. L. O., **COSTA, G. C. B.**, SIRQUEIRA, T. F. M., BRAGA, R., WERNER, C. M., CAMPOS, F., DAVID, J. M. N. "Using Ontology and Data Provenance to Improve Software Processes", In: Proceedings of the Brazilian Seminar on Ontologies (ONTOBRAS), pp. 10-21, 2015.

- **COSTA, G. C. B.** Using Data Provenance to Improve Software Process Enactment, Monitoring, and Analysis. In: Doctoral Symposium of IEEE/ACM International Conference on Software Engineering Companion (ICSE), IEEE, pp. 875-878, 2016.
- **COSTA, G. C. B.**, SCHOTS, M., OLIVEIRA, W. E. B., DALPRA, H. L. O., WERNER, C. M. L., BRAGA, R., DAVID, J. M. N., MIGUEL, M. A., STROELE, V., CAMPOS, F. SPPV: Visualizing Software Process Provenance Data. In: IV Workshop on Software Visualization, Evolution and Maintenance - VII Congresso Brasileiro de Software: Teoria e Prática (CBSoft 2016), 2016, Maringá. 4th Workshop on Software Visualization, Maintenance and Evolution (VEM 2016), pp. 49-56, 2016.
- **COSTA, G. C. B.**, WERNER, C. M., BRAGA, R. Software Process Performance Improvement Using Data Provenance and Ontology. In: International Conference on Business Process Management. Springer International Publishing, pp. 55-71, 2016.

Other works not directly related to the scope of this research:

- **COSTA, G.**, SILVA, F., SANTOS, R., WERNER, C., OLIVEIRA, T. From applications to a software ecosystem platform. In: Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems (MEDES), Neumunster Abbey, Luxembourg, pp. 9-16, 2013.
- **COSTA, G. C. B.**, SANTANA, F., MAGDALENO, A. M., WERNER, C. M. L. Monitoring Collaboration in Software Processes Using Social Networks. In: Baloian N., Burstein F., Ogata H., Santoro F., Zurita G. (eds) Collaboration and Technology. CRIWG 2014. Lecture Notes in Computer Science, vol 8658, pp.89-96, 2014.
- ORNELAS, T., BRAGA, R., DAVID, J. M. N., CAMPOS, F., **COSTA, G. C. B**. Provenance Data Discovery Through Semantic Web Resources. Concurrency and Computation - Practice and Experience, v.30, n.6, p. e.4366, 2018.

The following papers were submitted and were under revision until the completion of the thesis writing:

- **COSTA, G.**, WERNER, C., BRAGA, R., DALPRA, H., ARAÚJO, M., STROELE, V. Deriving Strategical Information for Software Development Processes using Provenance Data and Ontology Techniques. International Journal of Business Process Integration and Management. Submission: May 2018.

The following paper has just been accepted, and it is to be published:

- **COSTA, G.**, DALPRA, H., TEIXEIRA, E., WERNER, C., BRAGA, R., MIGUEL, M. Software Processes Analysis with Provenance. In: 19th International Conference on Product-Focuses Software Process Improvement (PROFES), Wolfsburg, Germany. Nov 2018. To appear.

8.3 Open Questions and Future Work

Considering PROV-SwProcess model, the following points should be considered as future work:

- Check PROV-SwProcess compliance using ProvValidator[30];
- Submit PROV-SwProcess for W3C;
- Make a detailed comparison with all PROV-SwProcess constructs and other PROV extensions;
- Explore other possibilities of stakeholder relationships (not just *acted on behalf of*) that could be captured during SDP, e.g. collaborative relationships between two or more stakeholders;
- PROV-SwProcess is divided into three levels (standard process, intended process and executed process) and, as future work, we can mention the possibility of deriving relationships that can be established and / or inferred across these levels. It was not addressed in its current version (besides the inference *HadRole*);
- Other possible improvement for PROV-SwProcess would be to address the relationships proposed in the Versioned-PROV (PIMENTEL *et al*., 2018), in order to deal with fine-grained provenance of software process artifacts and procedures;

[30] https://openprovenance.org/services/view/validator

- The process SDP3, presented in Chapter 6, had data from various projects and we just used one of them. We also consider as a future improvement to allow the interlacing of data from various projects, considering that it can bring important knowledge, including helping companies to migrate to a distributed development approach (e.g., ECOs); and

About PROV-SwProcess competency questions, the following points should be considered:

- Include / consider activities' time spent and their planned complexity in the analysis that considers the process activities; and
- Analyze the 'failure rate' of the development activities performed (if data related to the test activities were provided).

Several improvement points about iSPuP approach and its respective tool support raised after its evaluation with the process managers:

- Include the possibility of analyzing the data using queries in iSPuP tool support;
- Create a specific symbol for roles representation;
- Allow filtering by activity types;
- Implement a visualization that displays the relationship between each process instance, the product versions associated with it, and the affected stakeholders (*clients*) on each process instance;
- Include time filters in the analysis (using a slider to be aware of the changes in the process throughout the instances execution);
- Provide mechanisms for exporting the displayed data (in spreadsheet format); and
- Include filter replication between the two proposed views – graph and tabular (the same filter, when applied, should serve both views).

REFERENCES

ALAWINI, A., CHEN, L., DAVIDSON, S., FISHER, S., KIM, J. Discovering Similar Workflows via Provenance Clustering: a Case Study. In: 7th International Provenance and Annotation Workshop, London, United Kingdom, pp.115-127, 2018.

ACUNA, S. T., DE ANTONIO, A., FERRE, X., LOPEZ, M., MATE, L., ESTERO, S., "The Software Process: Modelling, Evaluation and Improvement", *Handbook of Software Engineering and Knowledge Engineering*, pp. 1-35, 2000.

BARRETO, A. S. "A Reuse-Based Approach to Define Processes Aiming at Processes High Maturity" [Uma Abordagem para Definição de Processos baseada em Reutilização Visando à Alta Maturidade em Processos] (in Portuguese). D.Sc. Thesis, COPPE/UFRJ, Rio de Janeiro, Brazil, 2011.

BASILI, V., CALDEIRA, G., ROMBACH, H. D. Goal Question Metric Paradigm. Encyclopedia of Software Engineering, v. 1, edited by John J. Marciniak, John Wiley & Sons, pp. 528–532, 1994.

BASKERVILLE, R., PRIES-HEJE, J., VENABLE, J. Soft design science methodology. In Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST '09). ACM, New York, NY, USA, Article 9, 11 pages, 2009.

BENDRAOU, R.; GERVAIS, M.-P. A Framework for Classifying and Comparing Process Technology Domains. In: International Conference on Software Engineering Advances (ICSEA 2007). IEEE, pp. 5, 2007.

BERARDI, R. C. G., RUIZ, D. Evaluating Data Quality of Software Effort: A Data Provenance Framework Based on Fuzzy-Logic. In: International Conference on Information Quality, Cambridge, Massachusetts. Proceedings of the 13th International Conference on Information Quality, v. 1. pp. 46-46, 2008.

BHATIA, M. P. S., KUMAR, A., BENIWAL, R. Ontologies for software engineering: Past, present and future. Indian Journal of Science and Technology, v. 9, n. 9, 2016.

BHATTACHARYA, P. Quantitative Decision-making in Software Engineering. D.Sc. Thesis, University of California, Riverside, United States. 2012.

BIRD C., MURPHY B., NAGAPPAN N., ZIMMERMANN T. Empirical Software Engineering at Microsoft Research. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work, Hangzhou, China, pp. 143-150, 2011.

BOSE, R., FREW, J. Lineage Retrieval for Scientific Data Processing: A Survey. ACM Computing Surveys, issue 1, vol. 37, pp. 1-28, 2005.

BOSCH, J. Speed, Data, and Ecosystems: Excelling in a Software-Driven World. CRC Press, 2017.

BRERETON, P., KITCHENHAM, B.A., BUDGEN, D., TURNER, M., KHALIL, M. Lessons from applying the systematic literature review process within the software engineering domain. Journal of systems and software, v. 80, n. 4, pp. 571-583, 2007.

BUNEMAN, P., KHANNA, S., TAN, W.C. Why and where: A characterization of data provenance. In: 8th International Conference on Database Theory, London. pp. 4-6, 2001.

BUSE, R. P., ZIMMERMANN, T. Information needs for software development analytics. In: Proceedings of the 34th International Conference on Software Engineering. IEEE Press, pp. 987-996, 2012.

COSTA, C., MURTA, L. Version control in distributed software development: A systematic mapping study. In: 8th International Conference on Global Software Engineering (ICGSE), IEEE, pp. 90-99, 2013.

COSTA, G. C. B. Using Data Provenance to Improve Software Process Enactment, Monitoring, and Analysis. In: IEEE/ACM International Conference on Software Engineering Companion (ICSE), IEEE, pp. 875-878, 2016.

COSTA, G. C. B., SCHOTS, M., OLIVEIRA, W. E. B., DALPRA, H. L. O., WERNER, C. M. L., BRAGA, R., DAVID, J. M. N., MIGUEL, M. A., STROELE, V., CAMPOS, F. SPPV: Visualizing Software Process Provenance Data. In: IV Workshop on Software Visualization, Evolution and Maintenance - VII Congresso Brasileiro de Software: Teoria e Prática (CBSoft 2016), 2016, Maringá. 4th Workshop on Software Visualization, Maintenance and Evolution (VEM 2016), pp. 49-56, 2016a.

COSTA, G. C. B., WERNER, C. M., BRAGA, R. Software Process Performance Improvement Using Data Provenance and Ontology. In: International Conference on Business Process Management. Springer International Publishing, pp. 55-71, 2016b.

CRUZ, S. M. S., CAMPOS, M. L. M., MATTOSO, M. Towards a Taxonomy of Provenance in Scientific Workflow Management Systems. In: Proceedings of the SERVICES '09 Congress on Services, pp. 259-266. Los Angeles, California, 2009.

CUEVAS-VICENTTÍN, V., LUDÄSCHER, B., MISSIER, P., BELHAJJAME, K., CHIRIGATI, F., WEI, Y., LEINFELDER, B. "ProvONE: A PROV Extension Data Model for Scientific Workflow Provenance". 2016. Available at: <https://purl.dataone.org/provone-v1-dev>. Accessed on: fev 2018.

DANG, Y. B., CHENG, P., LUO, L., CHO, A. A code provenance management tool for IP-aware software development. In: Companion of the 30th International Conference on Software Engineering, Informal Research Demonstrations. ACM. pp. 975-976, 2008.

DALPRA, H. L. O. "PROV-Process: Data Provenance Applied to Software Development Processes" [PROV-Process: Proveniência de Dados Aplicada a Processos de Desenvolvimento de Software] (in Portuguese). M.Sc. Dissertation, PGCC/UFJF, Juiz de Fora, Brazil, 2016.

DALPRA, H. L. O., COSTA, G., SIRQUEIRA, T. F. M., BRAGA, R., WERNER, C. M., CAMPOS, F., DAVID, J. M. N. "Using Ontology and Data Provenance to Improve Software Processes", In: Proceedings of the Brazilian Seminar on Ontologies, pp. 10-21, 2015.

DATA ONE. Data Observation Network of Earth, 2018 Available at: <https://www.dataone.org/>. Accessed on: fev 2018.

DAVIES, J., GERMAN, D. M., GODFREY, M. W., HINDLE, A. Software bertillonage: finding the provenance of an entity. Empirical Software Engineering, v. 18, n. 6, pp. 1195-1237, 2013.

DAVIDSON, S. B., BOULAKIA, S. C., EYAL, A., LUDÄSCHER, B., MCPHILLIPS, T. M., BOWERS, S., ANAND, M. K., FREIRE, J. Provenance in scientific workflow systems. IEEE Data Eng. Bull. v. 30, n. 4, pp. 44-50, 2007.

DAVIDSON, S. B., FREIRE, J. Provenance and scientific workflows: challenges and opportunities. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, pp. 1345-1350, 2008.

DE MELLO, R., MOTTA, R., TRAVASSOS, G. A Checklist-Based Inspection Technique for Business Process Models. In: International Conference on Business Process Management. Springer International Publishing, pp. 108-123, 2016.

DERNIAME, J. C., KABA, B. A., WASTELL, D. “Software Process: Principles, Methodology and Technology”. Berlin: Springer-Verlag, Lecture Notes in Computer Science, London, UK, 1999.

DE NIES, T. “Constraints of the PROV Data Model”, 2013. Available at: <https://www.w3.org/TR/prov-constraints/>. Accessed on: aug 2018.

FALBO, R. A., BERTOLLO, G. “A software process ontology as a common vocabulary about software processes”. International Journal of Business Process Integration and Management, v. 4, n. 4, pp. 239-250, 2009.

FALCI M. L., BRAGA R., STRÖELE V., DAVID, J. M. N. “Software Process Improvement through the Combination of Data Provenance, Ontologies and Complex Networks”, In: Proceedings of the 20th International Conference on Enterprise Information Systems, v. 2, pp. 61-70, 2018.

FREIRE, J., KOOP, D., SANTOS, E., SILVA, C. T. Provenance for Computational Tasks: A Survey. Computing in Science and Engineering, vol. 10, no. 3, pp. 11-21, 2008.

FUGGETTA, A. Software process: a roadmap. In: Proceedings of the Conference on The Future of Software Engineering, pp. 25-34, Limerick, Ireland, 2000.

FUGGETTA, A., DI NITTO, E. Software process. In: Proceedings of the on Future of Software Engineering. ACM, pp. 1-12, 2014.

GHOSHAL, D., PLALE, B. Provenance from log files: a BigData problem. In: Proceedings of the Joint EDBT/ICDT 2013 Workshops (EDBT '13). ACM, New York, NY, USA, pp. 290-297, 2013.

GIL, Y., MILES, S. “Prov model primer”, 2013. Available at: <http://www.w3.org/TR/prov-primer>. Accessed on: aug 2018.

GODFREY, M. W. Understanding software artifact provenance. Science of Computer Programming, v. 97, pp. 86-90, 2015.

GROTH, P. MOREAU, L. “PROV-Overview”, 2013. Available at: <https://www.w3.org/TR/prov-overview/>. Accessed on: oct 2018.

GUARINO, N. Formal ontology in information systems. In Proceedings of the 1st International Conference (FOIS'98), Trento, Italy, pp.3-15, 1998.

HERSCHEL, M., DIESTELKÄMPER, R., BEN LAHMAR, H. A survey on provenance: What for? What form? What from?. The VLDB Journal—The International Journal on Very Large Data Bases, v. 26, n. 6, pp. 881-906, 2017.

HEVNER, A., SALVATORE, T. M. JINSOO, P., SUDHA, R. Design science in information systems research. In: MIS Quarterly, vol 28, pp.75-105, 2004.

HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOF, B., DEAN, M. "SWRL: A semantic web rule language combining OWL and RuleML", 2004. Available at: <https://www.w3.org/Submission/SWRL/>. Accessed on: may. 2018.

HUMPHREY, W. S. Managing the Software Process. Boston, MA, USA, Addison-Wesley, 1989.

JABREF. JabRef reference manager. 2018. Available at: < http://jabref.sourceforge.net/>. Accessed on: fev 2018.

JØRGENSEN, H.D. "Software Process Model Reuse and Learning". In: Process Support for Distributed Team-based Software Development, Orlando, USA, pp. 726-731, 2000.

KITCHENHAM, B., CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report. 2007. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.471>. Accessed on: fev 2018.

LEAL, A. L. C., BRAGA, J. L., DA CRUZ, S. M. S. Cataloguing provenance-awareness with patterns. In: Proceedings of the 2015 IEEE Fifth International Workshop on Requirements Patterns (RePa). IEEE Computer Society, pp. 9-16, 2015.

LEBO, T., SAHOO, S., MCGUINNESS, D. "PROV-O: The PROV Ontology", 2013. Available at: <http://www.w3.org/TR/prov-o>. Accessed on: aug 2018.

LIM, C., LU, S., CHEBOTKO, A., FOTOUHI, F, "Prospective and Retrospective Provenance Collection in Scientific Workflow Environments". In: Proceedings of the 2010 IEEE International Conference on Services Computing (SCC '10). IEEE Computer Society, Washington, DC, USA, pp. 449-456, 2010.

LONCHAMP, J. A structured conceptual and terminological framework for software process engineering. In: Conference on the Software Process. IEEE Comput. Soc. Press, pp. 41-53, 1993.

MCAFEE, A., BRYNJOLFSSON, E. Big data: the management revolution. Harvard business review, v. 90, n. 10, p. 60-68, 2012.

MENZIES, T. ZIMMERMANN, T. Software analytics: so what?. IEEE Software, n. 4, pp. 31-37, 2013.

MILES, S. Automatically Adapting Source Code to Document Provenance. In: McGuinness D.L., Michaelis J.R., Moreau L. (eds) Provenance and Annotation of Data and Processes. IPAW 2010. Lecture Notes in Computer Science, v. 6378. Springer, Berlin, Heidelberg, pp. 102-110, 2010.

MILES, S., GROTH, P., MUNROE, S., MOREAU, L. PrIMe: A methodology for developing provenance-aware applications. ACM Transactions on Software Engineering and Methodology (TOSEM), v. 20, n. 3, 2011.

MISSIER, P., BELHAJJAME, K., CHENEY, J. The W3C PROV family of specifications for modelling provenance metadata. In: Proceedings of the 16th International Conference on Extending Database Technology. ACM, pp. 773-776, 2013a.

MISSIER, P., DEY, S., BELHAJJAME, K., CUEVAS-VICENTTIN, V., LUDAESCHER, B., D-prov: extending the prov provenance model with workflow structure. In: Proceedings of the 5th USENIX Workshop on the Theory and Practice of Provenance, TaPP 2013, pp. 9:1–9:7. 2013b.

MOUREAU, L. "The Foundations for Provenance on the Web". Foundations and Trends in Web Science, v.2, issue 2-3, Now Publishers, pp. 99-241, 2010.

MOREAU, L., CLIFFORD, B., FREIRE, J., FUTRELLE, J., GIL, Y., GROTH, P., KWASNIKOWSKA, N., MILES, S., MISSIER, P., MYERS, J. *et al*. The Open Provenance Model core specification (v1.1). Future Generation Computer Systems, v.27 n. 6, pp. 743 –756, 2011.

MOREAU, L., GROTH, P. Provenance: an introduction to prov. Synthesis Lectures on the Semantic Web: Theory and Technology, v. 3, n. 4, pp. 1-129, 2013.

MOREAU, L., LUDASCHER, B. ALTINTAS, I., BARGA, R.S., BOWERS, S., CHIN, G., COHEN, S., COHEN-BOULAKIA, S., CLIFFORD, B., DAVIDSON, S., DEELMAN, E., DIGIAMPIETRI, L., FOSTER, I., FREIRE, J., FREW, J., FUTRELLE, J., GIBSON, T., GIL, Y., GOBLE, C., GOLBECK, J., GROTH, P., HOLLAND, D. A., JIANG, S., KIM, J., KRENEK, A., MCPHILLIPS, T., MEHTA, G., MILES, S., METZGER, D., MUNROE, S., MYERS, J., PLALE, B., PODHORSZKI, N., RATNAKAR, V., SCHUCHARDT, K., SELTZER, M., SIMMHAN, Y. L., SLAUGHTER, P., STEPHAN, E., STEVENS, R., TURI, D., WILDE, M., ZHAO, J., ZHAO, Y. Special issue: The first provenance challenge.

Concurrency and computation: practice and experience, v. 20, n. 5, pp. 409-418, 2008.

MOREAU, L., MISSIER, P. “PROV-DM: The PROV Data Model”, 2013. Available at: <https://www.w3.org/TR/prov-dm/>. Accessed on: aug 2018.

MUNROE, S., MILES, S., MOREAU, L., V´AZQUEZ-SALCEDA, J. PrIMe: a software engineering methodology for developing provenance-aware applications. In: Proceedings of the 6th international workshop on Software engineering and middleware (SEM '06). ACM, NY, USA, pp. 39-46, 2006.

NGUYEN, M. N., CONRADI, R. “Classification of Meta-processes and their Models”. In: Proceedings of Third International Conference on the Applying Software Process, pp. 167-175, 1994.

NEIVA, F. W., DAVID, J. M. N., BRAGA, R., CAMPOS, F. Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature. *Information and Software Technology*, v. *72*, pp. 137-150, 2016.

OLIVEIRA, W., AMBRÓSIO, L. M., BRAGA, R., STRÖELE, V., DAVID, J. M., CAMPOS, F. A Framework for Provenance Analysis and Visualization. Procedia Computer Science, v. 108, pp. 1592-1601, 2017.

OLSSON, H. H., BOSCH, J. The HYPEX Model: From Opinions to Data-Driven Software Development. In: Continuous Software Engineering, Springer, Cham, pp. 155-164, 2014.

PAI, M., MCCULLOCH, M., GORMAN, J. D., PAI, N., ENANORIA, W., KENNEDY, G., THARYAN, P., COLFORD, J. M. Systematic Reviews and Meta-Analyses: An Illustrated, Step-by-Step Guide. The National Medical Journal of India, v. 17, n. 2, pp. 89-95, 2004.

PAULK, M. C. A history of the capability maturity model for software. ASQ Software Quality Professional, v. 12, n. 1, pp. 5-19, 2009.

PIMENTEL, J. F., MISSIER, P., MURTA, L., BRAGANHOLO, V. Versioned-PROV: A PROV extension to support mutable data entities. In: 7th International Provenance and Annotation Workshop, London, United Kingdom, pp.87-100, 2018.

RAM, S., LIU, J. Understanding the Semantics of Data Provenance to Support Active Conceptual Modeling. In: Active Conceptual Modeling of Learning. Chen, P.P., Wong, L.Y. (eds.). ACM-L 2006. Lecture Notes in Computer Science, v. 4512. Springer, Berlin, Heidelberg, pp. 17-29, 2007.

REIS, C.A., “A Flexible Approach to Evolvable Software Process Enactment” [Uma Abordagem Flexível para Execução de Processos de Software Evolutivos] (in Portuguese). D.Sc. Thesis, PPGC/UFRGS, Porto Alegre, Brazil, 2003.

RUY, F. B., FALBO, R. A., BARCELLOS, M. P., COSTA, S. D., GUIZZARDI, G. SEON: A software engineering ontology network. In: European Knowledge Acquisition Workshop. Springer, Cham, pp. 527-542, 2016.

SHULL, F., RUS I., BASILI, V. “How Perspective-Based Reading can Improve Requirements Inspections”, IEEE Computer, vol. 33, no. 7, pp. 73-79, 2000.

SIMMHAN, Y. L., PLALE, B., GANNON, D. A survey of data provenance in e-science. SIGMOD Rec. 34, 3 (September 2005), pp. 31-36, 2005.

SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A., KATZ, Y. Pellet: A practical owl-dl reasoner. Web Semantics: science, services and agents on the World Wide Web, v. 5, n. 2, pp. 51-53, 2007.

SUN, L., PARK, J., SANDHU, R. Engineering access control policies for provenance-aware systems. In: Proceedings of the third ACM conference on Data and application security and privacy. ACM, pp. 285-292, 2013.

TEIXEIRA, E. N., DE MELLO, R. M., MOTTA, R. C., WERNER, C. M. L., VASCONCELOS, A. Verification of software process line models: a checklist-based inspection approach. In: Proceedings of XVIII Ibero-American Conference on Software Engineering, Peru, Lima, 2015.

THAKUR, A., VAUGHN, R. ANANTHARAJ, V. Handling Undiscovered Vulnerabilities Using a Provenance Network. Journal of Systemics, Cybernetics and Informatics, v. 7, n. 3, p. 86-91, 2009.

TRAVASSOS, G. H., DOS SANTOS, P. S. M., MIAN, P. G., NETO, A. C. D., & BIOLCHINI, J. An Environment to Support Large Scale Experimentation in Software Engineering. In: 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2008), pp. 193-202, Belfast, March, 2008.

USCHOLD, M., GRUNINGER, M. Ontologies: Principles, methods and applications. The knowledge engineering review, v. 11, n. 2, pp. 93-136, 1996.

WENDEL, H., KUNDE, M., SCHREIBER, A. Provenance of software development processes. In: McGuinness D.L., Michaelis J.R., Moreau L. (eds) Provenance and Annotation of Data and Processes. IPAW 2010. Lecture Notes in Computer Science, vol 6378. Springer, Berlin, Heidelberg, pp. 59-63, 2010.

WOHLIN, C., RUNESON, P., HOST, M., OHLSSON, M. C., REGNELL, B., WESSLEN, A. Experimentation in software engineering. Springer, 2012.

WOLF, A. L., ROSENBLUM, D. S. A study in software process data capture and analysis. Software Process, 1993. In: Second International Conference on the Continuous Software Process Improvement. IEEE, pp. 115-124, 1993.

W3C, World Wide Web Consortium. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation. 2012. Available at: <http://www.w3.org/TR/owl2-overview/>. Accessed on: aug 2018.

YIN, R. K. Case Study Research Design and Methods. 5 ed. Beverly Hills: Sage Publications. 2014.

XU, P., SENGUPTA, A. Provenance in Software Engineering - A Configuration Management View. In: Proceedings of the Eleventh Americas Conference on Information Systems (AMCIS), Omaha, NE, USA, pp. 3103-3107, 2005.

APPENDIX A - SELECTED STUDIES

ID	Title and reference
1	A Code Provenance Management Tool for IP-Aware Software Development (Dang *et al*., 2008)
2	Cataloguing Provenance-Awareness with Patterns (Leal *et al*., 2015)
3	Engineering Access Control Policies for Provenance-aware Systems (Sun *et al*., 2013)
4	Evaluating Data Quality of Software Effort: A Data Provenance Framework Based on Fuzzy-Logic (Berardi and Ruiz, 2008)
5	Handling Undiscovered Vulnerabilities Using a Provenance Network (Thakur *et al*., 2009)
6	Provenance in Software Engineering - A Configuration Management View (Xu and Sengupta, 2005)
7	Provenance of software development processes (Wendel *et al*., 2010)
8	Software Bertillonage: Determining the provenance of software development artifacts (Davies *et al*., 2013)
9	Software process performance improvement using data provenance and ontology (Costa *et al*., 2016b)
10	Understanding software artifact provenance (Godfrey, 2015)
11	Understanding the semantics of data provenance to support active conceptual modeling (Ram and Liu, 2007)
12	Using Data Provenance to Improve Software Process Enactment, Monitoring, and Analysis (Costa, 2016)
13	Using ontology and data provenance to improve software processes (Dalpra *et al*., 2015)
14	Software Process Improvement through the Combination of Data Provenance, Ontologies and Complex Networks (Falci *et al*., 2018)

APPENDIX B - STUDIES EXTRACTION AND QUALITY FORMS

1. Dang *et al*., 2008.

Data	Extracted Data
Title of document:	A Code Provenance Management Tool for IP-Aware Software Development
Author(s):	Ya Bin Dang, Ping Cheng, Lin Luo, Adrian Cho
Publication date:	May/2008
Source:	International Conference on Software Engineering (ICSE)
Approach name:	Ariadne
Approach description:	A code provenance management tool is proposed. It tracks the provenance of source code and generate provenance reports to facilitate the management of its intellectual property (IP).
Provenance model:	The use of a specific provenance model is not mentioned. According to this study, *originality* information is divided into two types: editing history and IP-related information. 1. Editing history can be automatically generated by client monitoring. The types of editing events include insert a line, delete a line, modify a line, and copy-and-paste an object. 2. IP related information includes open-source claims, applicable patents, licensing terms, and contractual requirements. When it is first encountered, this information is entered by the developer through manual input, possibly after searching through source-code repositories. Upon reuse, the information is automatically combined with editing history information.
Approach benefits:	1. cost reduction of the copyright clearance effort 2. risk reduction of copyright contamination from external copy-and-paste
Artifacts:	source code
Provenance storage:	Metadata file with the same source code name, however, with different extension (*.orimeta).
Analysis method:	They analyze IP metadata to generate IP reports for the specified projects. These reports depict the everyday status of the project's IP pedigree, and project managers and attorneys can review the reports by browser or email. Unsafe items that violated the IP policies can be highlighted for proper actions.
Evaluation:	It is briefly mentioned (in just one paragraph of the text) that an evaluation of the proposal with three pilot projects was made, but it is not detailed.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Yes

QA3	Is the used provenance model clearly described and its adoption justified?	No
QA4	Is there any empirical/experimental result regarding the approach?	Partial
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	Partial

2. Leal *et al*., 2015.

Data	Extracted Data
Title of document:	Cataloguing Provenance-Awareness with Patterns
Author(s):	André Luiz de Castro Leal, José Luis Braga, Sérgio Manuel Serra da Cruz
Publication date:	Aug/2015
Source:	IEEE International Workshop on Requirements Patterns (RePa)
Approach name:	-
Approach description:	It is an approach to map provenance as a catalogue of non-functional requirement (NFR)
Provenance model:	This study proposed the modelling of provenance as a NFR catalogue.
Approach benefits:	Provenance is described as a Softgoal Interdependency Graph. The approach introduces patterns of provenance into the models of qualities of functional elements, describing it as a quality that can be satisfied to enhance the software, enabling the construction of chains of operations in software systems to produces pieces of data with higher quality.
Artifacts:	-
Provenance storage:	-
Analysis method:	SIG (Softgoal Interdependency Graph)
Evaluation:	To exemplify the use of Provenance SIG, a usage scenario in the scientific software domain is modelled.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Partial
QA3	Is the used provenance model clearly described and its adoption justified?	Partial
QA4	Is there any empirical/experimental result regarding the approach?	No
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	Partial

3. Sun *et al*., 2013.

Data	Extracted Data
Title of document:	Engineering Access Control Policies for Provenance-aware Systems
Author(s):	Lianshan Sun, Jaehong Park, Ravi Sandhu
Publication date:	Feb/2013
Source:	ACM conference on Data and application security and privacy (CODASPY)
Approach name:	-
Approach description:	The approach is a provenance-aware access control framework with a layered architecture that features an abstract layer, including a Typed Provenance Model (TPM). This model permits the identification, specification, and refinement of provenance-aware access control policies from the beginning of provenance-aware systems development.
Provenance model:	A Typed Provenance Model (TPM) based on OPM was described.
Approach benefits:	It permits to engineer provenance-aware access control policies from the beginning of provenance-aware systems development, using a TPM model that abstracts complex provenance graphs.
Artifacts:	Classes, business operations and actors
Provenance storage:	-
Analysis method:	-
Evaluation:	The paper illustrates the concept of TPM and its process implementations using a homework grading system.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Yes
QA3	Is the used provenance model clearly described and its adoption justified?	Yes
QA4	Is there any empirical/experimental result regarding the approach?	No
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	Yes

4. Berardi and Ruiz, 2008.

Data	Extracted Data
Title of document:	Evaluating Data Quality of Software Effort: A Data Provenance Framework Based on Fuzzy-Logic
Author(s):	Rita Cristina Galarraga Berardi, Duncan Ruiz
Publication date:	Nov/2008
Source:	International Conference on Information Quality (ICIQ)
Approach name:	-
Approach description:	A framework for evaluating software effort data is briefly described. It is divided in four major components: (1) Provenance Component: responsible for storing metadata

	traceability in a Provenance Database; (2) Inference Machine Component: represented by an inference machine that makes use of a previously created rules set based on fuzzy logic; (3) Quality Database Component: represented by a Quality Database that stores the output of the inference machine and (4) Provenance and Quality Warehouse Component: represented by a Data Warehouse that aims to provide analysis resources for the company management.
Provenance model:	-
Approach benefits:	Permits to the company to analyze the present state of the effort data, as well as to identify flawed points and improvement margins.
Artifacts:	-
Provenance storage:	The framework has a Provenance Database.
Analysis method:	-
Evaluation:	-

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	No
QA3	Is the used provenance model clearly described and its adoption justified?	No
QA4	Is there any empirical/experimental result regarding the approach?	No
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	No

5. Thakur *et al*., 2009.

Data	Extracted Data
Title of document:	Handling Undiscovered Vulnerabilities Using a Provenance Network
Author(s):	Amrit'anshu Thakur, Rayford Vaughn, Valentine Anantharaj
Publication date:	2009
Source:	Journal of Systemics, Cybernetics and Informatics
Approach name:	-
Approach description:	A method to address known and unknown vulnerabilities using concepts of provenance and pattern matching during the testing phase of a system's development lifecycle.
Provenance model:	No specific provenance data model is presented.
Approach benefits:	- A provenance-based trust network created during a systematic testing process is used as a reference point for the system's usage. This enables handling of known and unknown exceptions that could be potential threats to the system. - The final provenance network gives a quantified comparison of trust in a specific usage pattern.
Artifacts:	Program statements
Provenance storage:	It was not mentioned, but, through the text, it appears that a relational database was used to store the provenance data.

Analysis method:	Automated clustering based on individual cluster characteristics - put into place some form of clustering technique where 'most similar' candidates appear in the same group. This is performed in a mechanized fashion based on the attributes these candidates possess. Another step is manually aided interpolation to fully define a cluster's elements given its upper and lower bounds.
Evaluation:	A simple case study using real instances of input for a potential SQL injection attack is presented.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Yes
QA3	Is the used provenance model clearly described and its adoption justified?	Partial
QA4	Is there any empirical/experimental result regarding the approach?	No
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	No

6. Xu and Sengupta, 2005.

Data	Extracted Data
Title of document:	Provenance in Software Engineering - A Configuration Management View
Author(s):	Peng Xu, Arijit Sengupta
Publication date:	Aug/2005
Source:	Americas Conference on Information Systems (AMCIS)
Approach name:	The specific software configuration provenance model presented in the paper is called SCP Model and the approach prototype is called FTS (Fully Traceable System)
Approach description:	The approach presents how provenance can be achieved in configuration management by binding an artifact to its traceability and evolution information.
Provenance model:	SCP Model is used and described in the paper. It considers both traditional version control information and traceability among various artifacts across system lifecycle.
Approach benefits:	SCP model provides a new method to incorporate versioning, traceability, and provenance in software design. Such information is needed for many different applications, especially where software is developed in teams, where some teams may not have control over how other teams operate.
Artifacts:	Software development artifacts in general
Provenance storage:	XML-based metadata
Analysis method:	A component of FTS Architecture called "Inference engine" traces the dependency information in the XML file and suggest the impacted artifacts.
Evaluation:	-

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Yes
QA3	Is the used provenance model clearly described and its adoption justified?	Yes
QA4	Is there any empirical/experimental result regarding the approach?	No
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	Partial

7. Wendel *et al*., 2010.

Data	Extracted Data
Title of document:	Provenance of Software Development Processes
Author(s):	Heinrich Wendel, Markus Kunde, Andreas Schreiber
Publication date:	Jun/2010
Source:	International Provenance and Annotation Workshop (IPAW)
Approach name:	-
Approach description:	The paper presents an approach to make software development process (SDP) provenance-aware, using a service-oriented architecture to record/store provenance, PRiME (Munroe et al., 2006) and the Open Provenance Model. Its main goal is to answer questions related to the SDP, such as "Why does the build fail currently?".
Provenance model:	Open Provenance Model
Approach benefits:	They are not clearly presented; however, it has been inferred that the main benefits of the approach are record/store provenance data of software development process (using a high level of abstraction), allowing its querying.
Artifacts:	Interactions between developers in a distributed tool suite and the resulting artifacts
Provenance storage:	Graph database (Neo4j)
Analysis method:	Graph query language (Gremlin queries)
Evaluation:	It is only mentioned that the proposed approach has been implemented and evaluated using the software development process of a specific distributed simulation framework (Remote Computing Environment - http://rcenvironment.de/). The authors cited that the adapted methodology and selected technologies could be successfully used and offers the possibility to answer questions related to error detection, quality assurance, process validation, monitoring, statistical analysis, process optimization, developer rating and to informational purposes. It is presented that the approach showed a reasonable performance, however, this is not detailed / proven.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Partial

QA2	Is the presented approach clearly explained?	No
QA3	Is the used provenance model clearly described and its adoption justified?	No
QA4	Is there any empirical/experimental result regarding the approach?	No
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	No

8. Davies *et al*., 2013.

Data	Extracted Data
Title of document:	Software Bertillonage: Determining the Provenance of Software Development Artifacts
Author(s):	Julius Davies, Daniel M. German, Michael W. Godfrey, Abram Hindle
Publication date:	Dec/2013
Source:	Empirical Software Engineering
Approach name:	Software Bertillonage
Approach description:	This work has the following research question: "Given a software entity, can we determine where it came from, i.e., how can we establish its provenance?". It motivates the need for the recovery of the provenance of software entities by a broad set of techniques that could include signature matching, source code fact extraction, software clone detection, call ow graph matching, string matching, historical analyses, and other techniques. Given a library, file, function, or even snippet of code, this work determines the entity origin: "was the entity designed to fit into the design of the system where it sits, or has it been borrowed or adapted from another entity elsewhere?".
Provenance model:	-
Approach benefits:	When the provenance of software entities is determined, the stakeholders (software developers, IT managers, and the companies they work for) can use this information to comply with security standards, licensing, and other requirements.
Artifacts:	- software entities (a library, a file, a function, or even a snippet of code)
Provenance storage:	- a PostgreSQL database was used.
Analysis method:	It is used a technique of software Bertillonage: anchored signature matching. This method aids in reducing the search space when trying to determine the identity and version of a given Java archive within a large corpus of archives (such as the Maven 2 central repository).
Evaluation:	An empirical study on 945 jars from the Debian GNU/Linux distribution, as well as an industrial case study on 81 jars from an e-commerce application was conducted and explained to prove the validity of the proposed method.

ID	Quality assessment questions	Score

QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Yes
QA3	Is the used provenance model clearly described and its adoption justified?	No
QA4	Is there any empirical/experimental result regarding the approach?	Yes
QA5	Are threats to validity taken into consideration?	Yes
QA6	Are all research questions answered adequately?	Yes

9. Costa *et al*., 2016b.

Data	Extracted Data
Title of document:	Software Process Performance Improvement Using Data Provenance and Ontology
Author(s):	Gabriella Castro Barbosa Costa, Cláudia M. L. Werner, Regina Braga
Publication date:	Sep/2016
Source:	International Conference on Business Process Management – BPM 2016: Business Process Management Forum
Approach name:	-
Approach description:	An approach to support the reuse of experience in previous executions of software processes, using provenance data and ontology is proposed. This approach includes the software process enactment, monitoring and analysis improvement using provenance data and ontology and is divided into four distinct layers: (1) Client Layer: It is the interface between process members and the approach and allows the user's interaction and visualization of all process lifecycle; (2) Integration Layer: Integrates the Client Layer to all other layers of the approach, allowing the exchange of data/information between them; (3) Measure Layer: Is responsible for storing and capturing the measures related to the process to be executed and (4) Provenance Layer: Prospective and retrospective provenance data are captured, stored and imported into an ontology to make inferences using these data.
Provenance model:	PROV
Approach benefits:	This approach shows that using software process provenance data with ontologies we can provide implicit information to be used for improving process performance, using previously defined metrics. Using this approach, two specific types of information can be obtained: (1) Information related to the artifacts that are manipulated by the process, which helps to decrease runtime of new process instances; and (2) Information related to agents who already manipulated artifacts; thus, during the execution of a process, when a certain artifact is handled, the executor of the task could include new agents to the solution, given that they have used that artifact in some previous run and, therefore, could share some knowledge concerning it, which could possibly contribute to the reduction of the task

	runtime.
Artifacts:	Activities, entities, and agents
Provenance storage:	MySQL database
Analysis method:	The provenance analysis includes an ontology and an inference machine
Evaluation:	A pilot case study with data from two software development companies is presented to indicate the advantages of the proposed approach.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Yes
QA3	Is the used provenance model clearly described and its adoption justified?	Yes
QA4	Is there any empirical/experimental result regarding the approach?	Yes
QA5	Are threats to validity taken into consideration?	Yes
QA6	Are all research questions answered adequately?	Yes

10. Godfrey, 2015.

Data	Extracted Data
Title of document:	Understanding software artifact provenance
Author(s):	Michael W. Godfrey
Publication date:	Jan/2015
Source:	Science of Computer Programming
Approach name:	-
Approach description:	The paper analyses the problem of extracting and reasoning about the provenance of software development artifacts. The approach has two distinct phases: (1) a simple metric that is relatively cheap to compute on a large data set, is applicable at the level of granularity desired, and has good discriminatory value on candidates and (2) a more expensive and precise analysis on the result set from the first phase (e.g. an expensive clone detection algorithm might be used that requires deep static analysis of the code, or a manual analysis of the entities is done).
Provenance model:	PROV
Approach benefits:	The proposed approach of applying a computationally cheap and conceptually simple matching algorithm to a large data set, then applying a more expensive technique (a manual analysis of the best matches) worked well on the problem of matching library versions identifiers to a large space of possible matches taken from a near-comprehensive master repository
Artifacts:	software entities
Provenance storage:	A master database of well-known Java libraries using the Maven2 public repository as a basis is used.
Analysis method:	This approach uses a general two-phased strategy that is similar

	to the metaphor of Bertillonage (a 19th century approach to forensic analysis).
Evaluation:	An example of library version identification using the proposed approach is presented.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Yes
QA3	Is the used provenance model clearly described and its adoption justified?	Partial
QA4	Is there any empirical/experimental result regarding the approach?	No
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	Partial

11. Ram and Liu, 2007.

Data	Extracted Data
Title of document:	Understanding the Semantics of Data Provenance to Support Active Conceptual Modeling
Author(s):	Sudha Ram and Jun Liu
Publication date:	2007
Source:	International Workshop on Active Conceptual Modeling of Learning (ACM-L).
Approach name:	W7 model
Approach description:	An ontological model called W7 is presented and represents data provenance as a combination of seven interconnected elements including, “what”, “when”, “where”, “how”, “who”, “which”, and “why”. The semantics of each of these elements are presented in detail. These elements can be used to track provenance and may be applied to different domains.
Provenance model:	W7 model (proposed by the authors)
Approach benefits:	The main benefit of the approach is to present a generic model of data provenance and intends to be easily adaptable to represent domain or application specific provenance requirements in active conceptual modeling (requires capturing provenance knowledge in terms of what event/change may happen to the data, at the stage of conceptual modeling).
Artifacts:	Data objects at different granularity levels (e.g., instances of a class)
Provenance storage:	Provenance annotations
Analysis method:	-
Evaluation:	A homeland security example illustrates how current conceptual models can be extended to embed provenance.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes

QA2	Is the presented approach clearly explained?	Yes
QA3	Is the used provenance model clearly described and its adoption justified?	Yes
QA4	Is there any empirical/experimental result regarding the approach?	No
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	Yes

12. Costa, 2016.

Data	Extracted Data
Title of document:	Using Data Provenance to Improve Software Process Enactment, Monitoring, and Analysis
Author(s):	Gabriella Castro Barbosa Costa
Publication date:	May/2016
Source:	International Conference on Software Engineering (ICSE)
Approach name:	iSPuP (improving Software Process using Provenance)
Approach description:	The approach supports measurement definition, execution, monitoring, and analysis of software processes, to improve its performance by using provenance data, ontology, and predefined metrics.
Provenance model:	PROV
Approach benefits:	- Detection of artifacts that consume more process time, and provide two suggestions of how to decrease this runtime: (1) this artifact should be marked with the information that, if handled, may result in a process runtime increase in future process instances executions; (2) suggestions of agents who already manipulated this artifact in some previous run and, therefore, could share some knowledge concerning it, which could possibly contribute to the reduction of the task runtime. - Other contributions are cited (but not all are proved) as expected from this approach: (1) Provide support to the software process manager to define process metrics: these metrics will be collected and stored during the process execution phase and used to obtain information on how to improve software process as a whole; (2) Provide mechanisms for capturing software process prospective and retrospective provenance; (3) Provide mechanisms of feedback about possible improvements and adjustments to do in the defined process, based on process provenance data and measurements collected during process execution; (4) Provide mechanisms for visualizing process provenance data during the execution, monitoring and analysis phases; (5) Provide mechanisms for deriving implicit information related to process provenance data using ontology and inference machines; (6) Assessment of the feasibility and application / use of iSPuP approach using experimental studies.
Artifacts:	entities, activities, and agents
Provenance storage:	relational repository

Analysis method:	PROV-Process Ontology
Evaluation:	A pilot case study has been conducted to evaluate the proposed approach, considering software processes used in two real software development companies.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Partial
QA3	Is the used provenance model clearly described and its adoption justified?	Yes
QA4	Is there any empirical/experimental result regarding the approach?	Yes
QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	Partial

13. Dalpra *et al*., 2015.

Data	Extracted Data
Title of document:	Using Ontology and Data Provenance to Improve Software Processes
Author(s):	Humberto L. O. Dalpra, Gabriella C. B. Costa, Tássio F. M. Sirqueira, Regina Braga, Cláudia M. L. Werner, Fernanda Campos, José Maria N. David
Publication date:	Sep/2015
Source:	Brazilian Ontology Research Seminar (ONTOBRAS)
Approach name:	PROV-Process
Approach description:	The approach allows the storage and analysis of software process provenance data to identify improvements for future executions of software process instances by using a provenance layer (comprising a database, an ontology, and mechanisms to manipulate these components).
Provenance model:	PROV
Approach benefits:	- Extract strategic information to the project manager enabling her/him to take decisions that can improve process performance. - Using the approach, it is possible to detect: (1) activities that influenced the generation of other activities; (2) agents that could be associated with the solution of the deployment task, considering that they already handled the artifacts involved in this task in any other execution of the process; (3) A list of activities in which an agent was involved, as well as the artifacts (entities) handled by her/him.
Artifacts:	Activities, entities, and agents
Provenance storage:	PROV-Process relational database
Analysis method:	PROV-Process ontology
Evaluation:	The approach was applied to a process from a Brazilian software development company.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Yes
QA3	Is the used provenance model clearly described and their adoption justified?	Yes
QA4	Is there any empirical/experimental result regarding the approach?	Yes
QA5	Are threats to validity taken into consideration?	Yes
QA6	Are all research questions answered adequately?	Yes

14. Falci *et al.*, 2018.

Data	Extracted Data
Title of document:	Software Process Improvement through the Combination of Data Provenance, Ontologies and Complex Networks
Author(s):	Maria Luiza Falci, Regina Braga, Victor Stroële, José Maria N. David
Publication date:	Mar/2018
Source:	International Conference on Enterprise Information Systems (ICEIS 2018)
Approach name:	OntoComplex
Approach description:	OntoComplex is an architecture that uses ontology, complex networks, and inferences to derive implicit knowledge from provenance data related to software process. The main goal of the architecture, as quoted in the paper, is: "use software process and its execution data analysis, to help managers to make decisions based on acquired knowledge to improve future executions".
Provenance model:	An extension of ProvONE (ProvONEExt)
Approach benefits:	-Derive useful strategic knowledge to software managers from software process data; -Assist software managers in extracting knowledge and making better strategic decisions about the process.
Artifacts:	Process, ProcessExec, Data, and User
Provenance storage:	As ontology individuals and using Neo4j3 database management system
Analysis method:	Complex network analysis and ontological analysis
Evaluation:	An evaluation of the architecture using data from a medium-size company was presented, with brief descriptions about possible analysis, showing initial evidences of the architecture utility.

ID	Quality assessment questions	Score
QA1	Is the aim of the research sufficiently explained?	Yes
QA2	Is the presented approach clearly explained?	Partial
QA3	Is the used provenance model clearly described and their adoption justified?	Yes
QA4	Is there any empirical/experimental result regarding the approach?	No

QA5	Are threats to validity taken into consideration?	No
QA6	Are all research questions answered adequately?	Partial

APPENDIX C - PROV-SwProcess DISCREPANT CASES

PROV-SwProcess uses associations between classes to represent both prospective and retrospective provenance. An association occurs between two classes, always having a source (domain) and a destination (range). Associations DCs are listed from **DC.01** to **DC.22**.

A. Software Process Associations

A *Software Process* represents a software development process in its entirety and has the following associations: ***wasAttributedTo***, ***wasComposedBy*** and ***wasDerivedFrom*** to capture retrospective provenance and the association ***isComposedBy*** to capture prospective provenance.

Omission

DC.01 Some association needed to describe a performed software process (in addition to ***wasAttributedTo***, ***wasComposedBy*** and ***wasDerivedFrom***) was omitted from the model.

DC.02 Some association needed to describe the prospective provenance of a software process (in addition to ***isComposedBy***) was omitted from the model.

Incorrect fact

DC.03 Some software process association is not compliant with software development process.

Inconsistency

DC.04 Some software process association has the same semantic meaning (is duplicated in the model).

Ambiguity

DC.05 Some software process association is not clearly described, using ambiguous terms.

Extraneous information

DC.06 Some software process association does not belong to the provenance of software development process.

B. Activity Associations

An Activity represents a computational task in the software development process. It can be atomic or composed by other sub-activities.

Omission

DC.07 Some association needed to describe the activities that were performed in a software development process (in addition to ***wasAssociatedWith***, ***hadSubActivity***, ***wasInformedBy***, ***adopted***, ***changed***, ***used***, ***startedAtTime***, ***endedAtTime***) was omitted from the model.

DC.08 Some association needed to describe the activities to be executed in a software development process (in addition to ***precedes***, ***dependsOn***, ***hasSubActivity***) was omitted from the model.

Incorrect fact

DC.09 Some activity association is not compliant with software development process.

Inconsistency

DC.10 Some activity association has the same semantic meaning (is duplicated in the model).

Ambiguity

DC.11 Some activity association is not clearly described, using ambiguous terms.

Extraneous information

DC.12 Some activity association does not belong to the provenance of software development process.

C. Stakeholder Associations

A Stakeholder represents an agent involved, interested, or affected by the software process activities. It can be specialized in other four types: (i) Organization Stakeholder, (ii) Person Stakeholder, (iii) Project Stakeholder, and (iv) Team Stakeholder.

Omission

DC.13 Some association needed to describe the relation between stakeholders in a software development process (in addition to ***actedOnBehalfOf***) was omitted from the model.

Incorrect fact

DC.14 The stakeholder association is not compliant with software development process.

Inconsistency

DC.15 The stakeholder association has the same semantic meaning (is duplicated in the model) of another model association.

Ambiguity

DC.16 The stakeholder association is not clearly described, using ambiguous terms.

Extraneous information

DC.17 The stakeholder association does not belong to the provenance of software development process.

D. Artifact Associations

Artifacts represent the objects produced, changed, or used in the software development process activities. Artifacts can be of five types: (i) Software_Product, (ii) Software_Item, (iii) Document, (iv) Model, and (v) Information_Item.

Omission

DC.18 Some association whose origin is a software process artifact (in addition to ***wasGeneratedBy***, ***wasAttributedTo*** and ***wasDerivedFrom***) was omitted from the model.

Incorrect fact

DC.19 Some artifact association is not compliant with software development process.

Inconsistency

DC.20 Some artifact association has the same semantic meaning (is duplicated in the model).

Ambiguity

DC.21 Some artifact association is not clearly described, using ambiguous terms.

Extraneous information

DC.22 Some artifact association does not belong to the provenance of software development process.

Classes Discrepant cases

PROV-SwProcess has 20 classes to represent the provenance of Software Development Process. These classes are divided into five specific aspects, as shown in Table C.1. Classes DCs are listed from **DC.23** to **DC.27**.

Table C.1. PROV-SwProcess Classes

PROV-SwProcess Aspect	**Class Name**
Activity	Activity
	Software_Process
Stakeholder	Stakeholder
	Organization_Stakeholder
	Person_Stakeholder
	Project_Stakeholder
	Team_Stakeholder
Resource	Resource
	Software_Resource
	Hardware_Resource
Procedure	Procedure
	Method
	Document_Template
	Technique
Artifact	Artifact
	Software_Product
	Software_Item
	Document
	Model
	Information_Item

Omission

DC.23 Some class needed to describe the provenance of software development process (in addition to the classes in Table C.1), was omitted from the model.

Incorrect fact

DC.24 Some class is not compliant with software development process.

Inconsistency

DC.25 Some class has the same semantic meaning (is duplicated in the model).

Ambiguity

DC.26 Some class is not clearly described, using ambiguous terms.

Extraneous information

DC.27 Some class does not belong to the modeling software process domain.

Inferences Discrepant cases

PROV-SwProcess has six specific inference rules. Considering an inference as a rule that can be applied to PROV-SwProcess instances to add new PROV-SwProcess statements, **DC.28** to **DC.32** are created to deal with these inference rules.

Omission

DC.28 Some inference rule needed to describe the provenance of software development process was omitted from the model.

Incorrect fact

DC.29 Some inference rule is not compliant with software development process.

Inconsistency

DC.30 Some inference rule has the same semantic meaning (is duplicated in the model).

Ambiguity

DC.31 Some inference rule is not clearly described, using ambiguous terms.

Extraneous information

DC.32 Some inference rule does not belong to the modeling software process domain.

APPENDIX D - SUBJECT CHARACTERIZATION FORM

Name: _____

1. Academic degree:

[ ] Ph.D. Degree

[ ] Ph.D. Student

[ ] Master Degree

[ ] Master Student

[ ] Bachelor Degree

[ ] Undergraduate Student

2. Your current occupation is in:

[ ] Academia - Time (in years): _____

[ ] Industry - Time (in years): _____

[ ] Academia and Industry - Time (in years): _____

3. Please fill out your level of experience with SOFTWARE PROCESSES.

Please check all the options that apply.

[ ] None (if you choose this option, please do not choose any other one)

[ ] I have a superficial knowledge about this topic

[ ] I have a good knowledge about this topic

[ ] I studied this topic in a course/discipline

[ ] I studied this topic by reading one or more books

[ ] I used my knowledge about this topic in the context of a course in practice

[ ] I used my knowledge about this topic in personal projects

[ ] I used my knowledge about this topic in industry projects

4. Please fill out your level of experience with PROVENANCE.

Please check all the options that apply.

[ ] None (if you choose this option, please do not choose any other one)

[ ] I have a superficial knowledge about this topic

[ ] I have a good knowledge about this topic

[ ] I studied this topic in a course/discipline

[ ] I studied this topic by reading one or more books

[ ] I used my knowledge about this topic in the context of a course in practice

[ ] I used my knowledge about this topic in personal projects

[ ] I used my knowledge about this topic in industry projects

5. Please fill out your level of experience with PROVENANCE MODELS.

Please check all the options that apply.

[ ] None (if you choose this option, please do not choose any other one)

[ ] I have a superficial knowledge about this topic

[ ] I have a good knowledge about this topic

[ ] I studied this topic in a course/discipline

[ ] I studied this topic by reading one or more books

[ ] I used my knowledge about this topic in the context of a course in practice

[ ] I used my knowledge about this topic in personal projects

[ ] I used my knowledge about this topic in industry projects

6. Please fill out your level of experience with the PROV Model.

Please check all the options that apply.

[ ] None (if you choose this option, please do not choose any other one)

[ ] I have a superficial knowledge about this topic

[ ] I have a good knowledge about this topic

[ ] I studied this topic in a course/discipline

[ ] I studied this topic by reading one or more books

[ ] I used my knowledge about this topic in the context of a course in practice

[ ] I used my knowledge about this topic in personal projects

[ ] I used my knowledge about this topic in industry projects

APPENDIX E - EVALUATION FORM (VERSION 1)

Name: _____

Instructions:

1. The model to be evaluated, is available at this link: http://www.gabriellacastro.com.br/provswprocess (*password*: modelodsc).
2. For each of the questions, we ask that one of the following be chosen:
   - ***Yes***
   - ***I don't know / I am not sure***
   - ***No***

   If the option '***Yes***' is chosen, we would like to receive some justification in order to analyze the problem and try to improve the proposed model.
3. Any comments regarding the evaluation or other comments about the model should be kept at the end of this form.

Thank you.

A. Software Process Associations

A *Software Process* represents a software development process in its entirety and has the following associations: ***wasAttributedTo***, ***wasComposedBy*** and ***wasDerivedFrom*** to capture retrospective provenance and the association ***isComposedBy*** to capture prospective provenance.

A-Q1) Is some association needed to describe a performed software process (in addition to ***wasAttributedTo***, ***wasComposedBy*** and ***wasDerivedFrom***) omitted from the model?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q2) Is some association needed to describe the prospective provenance of a software process (in addition to ***isComposedBy***) omitted from the model?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q3) Is some software process association not compliant with software development process?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q4) Has some software process association the same semantic meaning (are duplicated in the model)?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q5) Is some software process association not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q6) Does some software process association not belong to the provenance of software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B. Activity Associations

An Activity represents a computational task in the software development process. It can be atomic or composed by other sub-activities.

B-Q1) Is some association needed to describe the activities that were performed in a software development process (in addition to ***wasAssociatedWith***, ***hadSubActivity***, ***wasInformedBy***, ***adopted***, ***changed***, ***used***, ***startedAtTime***, ***endedAtTime***) omitted from the model?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q2) Is some association needed to describe the activities to be executed in a software development process (in addition to ***precedes***, ***dependsOn***, ***hasSubActivity***) omitted from the model?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q3) Is some activity association not compliant with software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q4) Has some activity association the same semantic meaning (is duplicated in the model)?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q5) Is some activity association not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q6) Does some activity association not belong to the provenance of software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C. Stakeholder Associations

A Stakeholder represents an agent involved, interested, or affected by the software process activities. It can be specialized in other four types: (i) Organization Stakeholder, (ii) Person Stakeholder, (iii) Project Stakeholder, and (iv) Team Stakeholder.

C-Q1) Is some association needed to describe the relation between stakeholders in a software development process (in addition to ***actedOnBehalfOf***) omitted from the model?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q2) Is the stakeholder association not compliant with software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q3) Has the stakeholder association the same semantic meaning (is duplicated in the model) of another model association?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q4) Is the stakeholder association not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q5) Does the stakeholder association not belong to the provenance of software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D. Artifact Associations

Artifacts represent the objects produced, changed, or used in the software development process activities. Artifacts can be of five types: (i) Software_Product, (ii) Software_Item, (iii) Document, (iv) Model, and (v) Information_Item.

D-Q1) Is some association whose origin is a software process artifact (in addition to ***wasGeneratedBy***, ***wasAttributedTo*** and ***wasDerivedFrom***) omitted from the model?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D-Q2) Is some artifact association not compliant with software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D-Q3) Has some artifact association the same semantic meaning (is duplicated in the model)?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D-Q4) Is some artifact association not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D-Q5) Does some artifact association not belong to the provenance of software development process?

[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E. Classes

PROV-SwProcess has 20 classes to represent the provenance of Software Development Process. These classes are divided into five specific aspects, as shown in Table 1.

Table 1. PROV-SwProcess Classes

PROV-SwProcess Aspect	Class Name
Activity	Activity
	Software_Process
Stakeholder	Stakeholder
	Organization_Stakeholder
	Person_Stakeholder
	Project_Stakeholder
	Team_Stakeholder
Resource	Resource
	Software_Resource
	Hardware_Resource
Procedure	Procedure
	Method
	Document_Template
	Technique
Artifact	Artifact
	Software_Product
	Software_Item
	Document
	Model
	Information_Item

E-Q1) Is some class needed to describe the provenance of software development process (in addition to the classes in Table 1), omitted from the model?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E-Q2) Is some class not compliant with software development process?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E-Q3) Has some class the same semantic meaning (is duplicated in the model)?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E-Q4) Is some class not clearly described, using ambiguous terms?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E-Q5) Does some class not belong to the modeling software process domain?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F. Inferences

PROV-SwProcess has six specific inference rules. An inference is a rule that can be applied to PROV-SwProcess instances to add new PROV-SwProcess statements.

F-Q1) Is some inference rule needed to describe the provenance of software development process omitted from the model?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F-Q2) Is some inference rule not compliant with software development process?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F-Q3) Has some inference rule the same semantic meaning (is duplicated in the model)?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F-Q4) Is some inference rule is not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F-Q5) Does some inference rule not belong to the modeling software process domain?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

General comments about the model:

_____

General comments about the evaluation:

_____

APPENDIX F - EVALUATION FORM (VERSION 2)

Name: _____

Instructions:

4. The model to be evaluated, is available at this link: http://www.gabriellacastro.com.br/provswprocess/v2.html

5. For each of the questions, we ask that one of the following be chosen:
   - ***Yes***
   - ***I don't know / I am not sure***
   - ***No***

   If the option '***Yes***' is chosen, we would like to receive some justification in order to analyze the problem and try to improve the proposed model.

6. Any comments regarding the evaluation or other comments about the model should be kept at the end of this form.

Thank you.

A. Software Process Associations

A *Software Process* represents a software development process in its entirety and has the following associations: ***wasAttributedTo*** and ***wasComposedBy*** to capture retrospective provenance and the associations ***hasResponsible*** and ***isComposedBy*** to capture prospective provenance.

A-Q1) Is some association needed to describe a performed software process (in addition to ***wasAttributedTo*** and ***wasComposedBy***) omitted from the model?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q2) Is some association needed to describe the prospective provenance of a software process (in addition to ***hasResponsible*** and ***isComposedBy***) omitted from the model?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q3) Is some software process association not compliant with software development process?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q4) Has some software process association the same semantic meaning (are duplicated in the model)?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q5) Is some software process association not clearly described, using ambiguous terms?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

A-Q6) Does some software process association not belong to the provenance of software development process?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B. Activity Associations

An Activity represents a computational task in the software development process. It can be atomic or composed by other sub-activities and may include the adoption of procedures, the use of resources, the modification, use and generation of artifacts, and the association with stakeholders responsible for its execution.

B-Q1) Is some association needed to describe the activities that were performed in a software development process (in addition to ***adopted, changed, generated, used, wasAssociatedWith, wasInformedBy, wasSubActivity, startedAtTime,*** and ***endedAtTime***) omitted from the model?

[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q2) Is some association needed to describe the activities to be executed in a software development process (in addition to ***adopts, changes, generates, isAssociatedWith, isSubActivity, precedes,*** and ***uses***) omitted from the model?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q3) Is some activity association not compliant with software development process?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q4) Has some activity association the same semantic meaning (is duplicated in the model)?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q5) Is some activity association not clearly described, using ambiguous terms?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

B-Q6) Does some activity association not belong to the provenance of software development process?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C. Stakeholder Associations

A Stakeholder represents an agent involved, interested, or affected by the software process activities. It can be specialized in other three types: (i) Organization Stakeholder, (ii) Person Stakeholder, and (iii) Team Stakeholder.

C-Q1) Is some association needed to describe the relations occurred between stakeholders and other software development process constructs (in addition to ***actedOnBehalfOf, created, modified,*** and ***hadRole***) omitted from the model?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q2) Is some association needed to describe the planned relations to occur between stakeholders and other software development process constructs (in addition to ***actsOnBehalfOf*** and ***hasRole***) omitted from the model?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q3) Is the stakeholder association not compliant with software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q4) Has the stakeholder association the same semantic meaning (is duplicated in the model) of another model association?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q5) Is the stakeholder association not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

C-Q6) Does the stakeholder association not belong to the provenance of software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D. Artifact Associations

Artifacts represent the objects produced, changed, or used in the software development process activities. Artifacts can be of five types: (i) Software_Product, (ii) Software_Item, (iii) Document, (iv) Model, and (v) Information_Item.

D-Q1) Is some occurred association whose origin is a software process artifact (in addition to ***wasBasedOn, wasDerivedFrom, generatedAtTime,*** and ***invalidatedAtTime***) omitted from the model?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D-Q2) Is some artifact association not compliant with software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D-Q3) Has some artifact association the same semantic meaning (is duplicated in the model)?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D-Q4) Is some artifact association not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

D-Q5) Does some artifact association not belong to the provenance of software development process?

[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E. Procedure Association

Procedures represent a normative description prescribing a defined way for performing the software development process activities. It can be of three types: (i) Method, (ii) Document Template, and (iii) Technique.

E-Q1) Is some occurred association whose origin is a procedure (in addition to ***wasAppliedTo***) omitted from the model?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E-Q2) Is the procedure association not compliant with software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E-Q3) Has the procedure association the same semantic meaning as other association (is duplicated in the model)?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E-Q4) Is the procedure association not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

E-Q5) Do the procedure association not belong to the provenance of software development process?

[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F. Classes

PROV-SwProcess has 20 classes to represent the provenance of software development process. These classes are divided into five specific aspects, as shown in Table 1.

Table 1. PROV-SwProcess Classes

PROV-SwProcess Aspect	Class Name
Activity	Activity
	Software_Process
Stakeholder	Stakeholder
	Organization_Stakeholder
	Person_Stakeholder
	Team_Stakeholder
	Stakeholder_Role
Resource	Resource
	Software_Resource
	Hardware_Resource
Procedure	Procedure
	Method
	Document_Template
	Technique
Artifact	Artifact
	Software_Product
	Software_Item
	Document
	Model
	Information_Item

F-Q1) Is some class needed to describe the provenance of software development process (in addition to the classes in Table 1), omitted from the model?

[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F-Q2) Is some class not compliant with software development process?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F-Q3) Has some class the same semantic meaning (is duplicated in the model)?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F-Q4) Is some class not clearly described, using ambiguous terms?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

F-Q5) Does some class not belong to the modeling software process domain?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

G. Inferences

PROV-SwProcess has seven groups of inferences (fifteen inference rules in total). An inference as a rule that can be applied to PROV-SwProcess instances to add new PROV-SwProcess statements.

G-Q1) Is some inference rule needed to describe the provenance of software development process omitted from the model?
[ ] *Yes* –
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

G-Q2) Is some inference rule not compliant with software development process?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

G-Q3) Has some inference rule the same semantic meaning (is duplicated in the model)?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

G-Q4) Is some inference rule is not clearly described, using ambiguous terms?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

G-Q5) Does some inference rule not belong to the modeling software process domain?
[ ] *Yes –*
Justification:_____

[ ] I *don't know / I am not sure*

[ ] *No*

General comments about the model:

General comments about the evaluation:

APPENDIX G - INTERVIEW SCRIPT WITH COMPANY MANAGERS

- **Explain the thesis main goal**: *Propose and evaluate an approach for capturing, storing, discovering and visualizing SDP execution provenance data to support process analysis and data-driven decision-making.*

- **Explain the interview goal**: analyze some *questions* (and *goals*) that the approach tries to answer, using the provided process execution data.

Company and Manager Characterization Questions

( ) Company 1 ( ) Company 2 ( ) Company 3

1. Job Title:
2. Experience as a manager (in years): (0 – 1 – 2 – 3…8 – 9 – 10 - More than 10)
3. Company Age:

( ) 9 years or less operating

( )10 year or more operating

4. Company Size:

( ) Micro: less 10 employees

( ) Small: among 10 and 49 employees

( ) Medium-size: among 50 and 99 employees

( ) Large: more than 100 employees

5. Brief description of the software development process(es) that you manage:

6. What tool(s) do you use to analyze and make decisions about the process?

Opinion Questions

1. Considering the question "**CQ1 - What are the process activities, artifacts, resources, procedures, stakeholders, and the relations among them during the process execution?**", the iSPuP approach gives the following views:

   - Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ1 - Analysis**: It is possible to identify all the process elements that participated in process executions and the relation among them.

   a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially
   b. Can this analysis assist in the following decision-making?

      CQ1 - Decision-Making Possibility: After identifying the process elements and the relation between them it is possible to find gaps (elements without association or inadequate relation established) in the analyzed data and try to correct it in next process executions.

      ( ) Yes ( ) No ( ) Partially
   c. Could you answer **CQ1** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially
   d. What is the relevance of answering **CQ1** to support in analysis and decision-making processes?

      ( ) Extremely relevant

      ( ) Very relevant

      ( ) Somewhat relevant

      ( ) Not very relevant

      ( ) Irrelevant

2. Considering the question "**CQ2 - Which procedures are used by the process during its execution?**", the iSPuP approach gives the following views:

   - Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ2 - Analysis**: It is possible to check which procedures influenced an artifact development; Verify the procedures most useful in the analyzed instance(s), when a procedure is used by artifacts in a number greater than the average; Check procedures useless, i.e., although existing, these procedures were never used during the execution of the processes carried out by the organization.

a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially

b. Can this analysis assist in the following decision-making?

CQ2 - Decision-Making Possibility: When verifying that procedures influenced an artifact development, the process manager can evaluate if this fact was really planned/expected (in process modeling phase) or not; if this information is not specified in the process model, the process manager may include it; Being aware that a procedure is widely used by the process instances, the manager can better plan any changes in this procedure, since this can have a great impact on future executions; If a procedure has not been used during process execution, this information may be valid for the process manager to evaluate whether this procedure needs to be changed/reshaped to be used as planned or if it should be removed from the process. Another point of analysis would be the impact of not having a standard for the development of some artifacts – it could impact the quality level of generated artifacts, as well as cause errors by the difficulty of understanding some information in these artifacts, etc.

( ) Yes ( ) No ( ) Partially

c. Could you answer **CQ2** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially

d. What is the relevance of answering **CQ2** to support in analysis and decision-making processes?

( ) Extremely relevant

( ) Very relevant

( ) Somewhat relevant

( ) Not very relevant

( ) Irrelevant

3. Considering the question "**CQ3 - Which activities had a high complexity (considering the number of associated stakeholders, artifacts, procedures and / or resources)?**", the iSPuP approach gives the following views:

- Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ3 - Analysis**: It is possible to check when activities are associated with many stakeholders, artifacts, procedures and / or

resources, when compared to the other activities of the process, indicating that this activity could be more complex than others.

a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially
b. Can this analysis assist in the following decision-making?

   CQ3 - Decision-Making Possibility: With the information provided by the analysis presented above, the process manager can evaluate if this fact was really planned/expected (in process modeling phase) or not; if this information is not specified in the process model, the process manager may change the process model to better represent the process that was in fact executed; A possible evaluation of the activities detected as more complex can be performed, aiming to divide it into less complex sub activities.

   ( ) Yes ( ) No ( ) Partially
c. Could you answer **CQ3** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially
d. What is the relevance of answering **CQ3** to support in analysis and decision-making processes?

   ( ) Extremely relevant
   ( ) Very relevant
   ( ) Somewhat relevant
   ( ) Not very relevant
   ( ) Irrelevant

4. Considering the question "**CQ4 - Which activities had a high dependency (on other activities)?**", the iSPuP approach gives the following views:

   - Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ4 - Analysis**: It is possible to analyze the dependency between two activities, i.e., when occurred the exchange of some artifact by two activities, one activity using some entity generated or changed by the other. It is also possible to discover which activity occurred before or after another during execution time and to identify possible bottlenecks based on activities dependency.

   a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially
   b. Can this analysis assist in the following decision-making?

CQ4 - Decision-Making Possibility: From the previous analyzes, the process manager can confront the activities (and its flow) specified in the process model and how they occurred during execution. If there are any discrepancies, he can make changes in the process model, according to what he verified that, in fact, was executed. Another decision is trying to make chances in the process model in order to avoid bottlenecks, if it were identified in the previous analysis.

( ) Yes ( ) No ( ) Partially

c. Could you answer **CQ4** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially

d. What is the relevance of answering **CQ4** to support in analysis and decision-making processes?

( ) Extremely relevant

( ) Very relevant

( ) Somewhat relevant

( ) Not very relevant

( ) Irrelevant

5. Considering the question "**CQ5 - What is the activities distribution among stakeholders?**", the iSPuP approach gives the following views:

- Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ5 - Analysis**: It is possible to discover, from a stakeholder, all the activities (and the total of these activities) in which he/she participated, allowing to understand the activities distribution among stakeholders in the process execution.

a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially

b. Can this analysis assist in the following decision-making?

CQ5 - Decision-Making Possibility: When verifying that a stakeholder is participating in much activities than others, the process manager can evaluate if this fact was really planned/expected (considering, for example, that a stakeholder was associated to a high number of activities because him/her always is attributed to activities with a lower level of complexity) or if it has been occurring due to an inadequate activity distribution during the process instantiation.

( ) Yes ( ) No ( ) Partially

c. Could you answer **CQ5** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially

d. What is the relevance of answering **CQ5** to support in analysis and decision-making processes?

( ) Extremely relevant

( ) Very relevant

( ) Somewhat relevant

( ) Not very relevant

( ) Irrelevant

6. Considering the question "**CQ6 - Which artifacts are known by a stakeholder, considering that in some process execution he/she created or modified such artifact?**", the iSPuP approach gives the following views:

- Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ6 - Analysis**: It is possible to discover all the artifacts that were created and / or modified by a stakeholder, allowing to understand about what artifacts this stakeholder has some knowledge, considering he/she manipulated this artifact in some process execution. Considering the artifact view point, it is possible to discover all the stakeholders that have some knowledge about it, considering it was created or modified by them.

a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially

b. Can this analysis assist in the following decision-making?

CQ6 - Decision-Making Possibility: in a future instantiation of the analyzed process, if a certain task is associated with a specific artifact, the process manager (or the responsible for the process instantiation) can allocate to this task a stakeholder with greater or less knowledge about the artifact to be manipulated during this task execution, according to the project objectives / goals.

( ) Yes ( ) No ( ) Partially

c. Could you answer **CQ6** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially

d. What is the relevance of answering **CQ6** to support in analysis and decision-making processes?

( ) Extremely relevant

( ) Very relevant

( ) Somewhat relevant

( ) Not very relevant

( ) Irrelevant

7. Considering the question "**CQ7 - Which stakeholders are out of the average of created and/or modified artifacts?**", the iSPuP approach gives the following views:

   - Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ7 - Analysis**: It is possible to discover, from a stakeholder, the total and which artifacts were created or modified by him/her, allowing to understand the performance of this stakeholder considering the manipulation of process artifacts (e.g., if he/she usually creates new artifacts or if he/she only modifies them).

   a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially
   b. Can this analysis assist in the following decision-making?

      CQ7 - Decision-Making Possibility: When verifying that a stakeholder is creating more artifacts than others, the process manager can evaluate if they really need to be created or if there is a stakeholder's lack of knowledge about the existing and available artifacts to be changed/adapted; - The process manager can better specify the responsible for the artifacts manipulation in a future instantiation of the analyzed process in order to obtain a better balance in relation to the stakeholder performance, considering the number of artifacts handled by the stakeholders.

      ( ) Yes ( ) No ( ) Partially
   c. Could you answer **CQ7** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially
   d. What is the relevance of answering **CQ7** to support in analysis and decision-making processes?

      ( ) Extremely relevant

      ( ) Very relevant

      ( ) Somewhat relevant

      ( ) Not very relevant

( ) Irrelevant

8. Considering the question "**CQ8 - What are the relationships among stakeholders?**", the iSPuP approach gives the following views:

   - Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ8 - Analysis**: It is possible to know the responsibility between the stakeholders during a process instance execution, detecting whether one stakeholder is responsible for many others or not.

   a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially
   b. Can this analysis assist in the following decision-making?

      CQ8 - Decision-Making Possibility: after analyzing the responsibility among stakeholders in executed instances, the process manager can use this information when allocating the responsibilities between stakeholders when a new instance of this process is created, according to the project objectives / goals.

      ( ) Yes ( ) No ( ) Partially
   c. Could you answer **CQ8** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially
   d. What is the relevance of answering **CQ8** to support in analysis and decision-making processes?

      ( ) Extremely relevant
      ( ) Very relevant
      ( ) Somewhat relevant
      ( ) Not very relevant
      ( ) Irrelevant

9. Considering the question "**CQ9 - Which roles each stakeholder assumes?**", the iSPuP approach gives the following views:

   - Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ9 - Analysis**: It is possible to analyze all the roles that have already been played by a specific stakeholder as well as, from a role, to verify which stakeholders can accomplish it.

   a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially
   b. Can this analysis assist in the following decision-making?

CQ9 - Decision-Making Possibility: In a next instantiation of this process, if the process manager needs to allocate some person stakeholder in a specific activity that needs some pre-defined role, he can evaluate who can perform this role, based on stakeholders' skills. On the other hand, he can also decide who should participate in a training programming to be able to accomplish more roles during process execution.

( ) Yes ( ) No ( ) Partially

c. Could you answer **CQ9** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially

d. What is the relevance of answering **CQ9** to support in analysis and decision-making processes?

( ) Extremely relevant

( ) Very relevant

( ) Somewhat relevant

( ) Not very relevant

( ) Irrelevant

10. Considering the question "**CQ10 - Which artifacts are derivations from others?**", the iSPuP approach gives the following views:

- Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ10 - Analysis**: It is possible to discover all the artifacts derived from others in addition to verify the artifacts that were most used for the derivation of others and, therefore, are of great importance in the analyzed SDP.

a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially

b. Can this analysis assist in the following decision-making?

CQ10 - Decision-Making Possibility: When verifying that an artifact was much used for the derivation of others, the changes in this artifact must be well planned to avoid that all the various other artifacts derived from it also need to be changed.

( ) Yes ( ) No ( ) Partially

c. Could you answer **CQ10** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially

d. What is the relevance of answering **CQ10** to support in analysis and decision-making processes?

( ) Extremely relevant

( ) Very relevant

( ) Somewhat relevant

( ) Not very relevant

( ) Irrelevant

11. Considering the question "**CQ11 - Which artifacts or procedures are revisions from others?**", the iSPuP approach gives the following views:

- Show to the participant the tool screen that displays these data (graph and table) and comment the following analysis: **CQ11 - Analysis**: It is possible to discover all the artifacts revisions, in addition to its latest versions / revisions.

a. Is this analysis correct? ( ) Yes ( ) No ( ) Partially

b. Can this analysis assist in the following decision-making?

CQ11 - Decision-Making Possibility: It is possible to evaluate when the last revision of a given artifact occurred, in addition to show if an artifact has already suffered many or no changes. This information can help in defining which artifact (or procedure) can / should be used in a next process execution.

( ) Yes ( ) No ( ) Partially

c. Could you answer **CQ7** using your current process management tool or dashboard? ( ) Yes ( ) No ( ) Partially

d. What is the relevance of answering **CQ7** to support in analysis and decision-making processes?

( ) Extremely relevant

( ) Very relevant

( ) Somewhat relevant

( ) Not very relevant

( ) Irrelevant

Final Questions

1. Are the questions adequate and sufficient to achieve their goals?

( ) Yes ( ) No ( ) Partially

Justification:

Goal 1: Process structure identification during execution and possibilities for process redesign

- CQ1 What are the process activities, artifacts, resources, procedures, stakeholders, and the relations among them during the process execution?
- CQ2: Which procedures are used by the process during its execution?
- CQ3: Which activities had a high complexity (considering the number of associated stakeholders, artifacts, procedures and / or resources)?
- CQ4: Which activities had a high dependency?

Goal 2: Understanding stakeholder's involvement in process execution

- CQ5: What is the activities distribution among stakeholders?
- CQ6: Which artifacts are known by a stakeholder, considering that in some process execution he/she created or modified such artifact?
- CQ7: Which stakeholders are out of the average of created and/or modified artifacts?
- CQ8: What are the relationships among stakeholders?
- CQ9: Which roles each stakeholder assumes?

Goal 3: Tracking derivations and revisions among artifacts or procedures

- CQ10: Which artifacts are derivations from others?
- CQ11: Which artifacts or procedures are revisions from others?

2. Do you suggest any other question that should be considered relevant to assist in SDP analysis and decision-making?

3. Do you have any comment about the interview?

4. Do you have any comment about the approach?

APPENDIX H - SDP2: DETAILED EXECUTION AND ANALYSIS

This Appendix presents a detailed discussion about all the PROV-SwProcess Competency Questions, using data from SDP2 and including the manager's opinion on each of them (the same procedure reported for SDP1 was followed, changing only the scenario and the subject).

Goal 1: Process structure identification during execution and possibilities for process redesign

– **CQ1** *What are the process activities, artifacts, resources, procedures, stakeholders, and the relations among them during the process execution?*

Considering the analysis possibility presented by CQ1 (*It is possible to identify all the process elements that participated in the process execution and the relation among them*) and its respective decision-making possibility (*After identifying the process elements and the relation between them it is possible to find gaps – elements without association or inadequate relation established*), when using iSPuP tool support and the generated macro visualization about SDP2 (shown on Figure 7.13), a grouping of four ellipses can be seen in the lower left corner of this figure. In addition, this group of four ellipses does not have any relation to the other process elements. By hovering over this grouping, the tooltip shown in Figure H.1 is displayed. Based on this, we consider this fact as a gap possibility: three roles informed in the process model were not associated with any of the stakeholders involved in the process (*Test*, *Development_Manager*, and *Support_Manager*) and, in addition, one of the artifacts generated in any of the instances did not have its name informed.

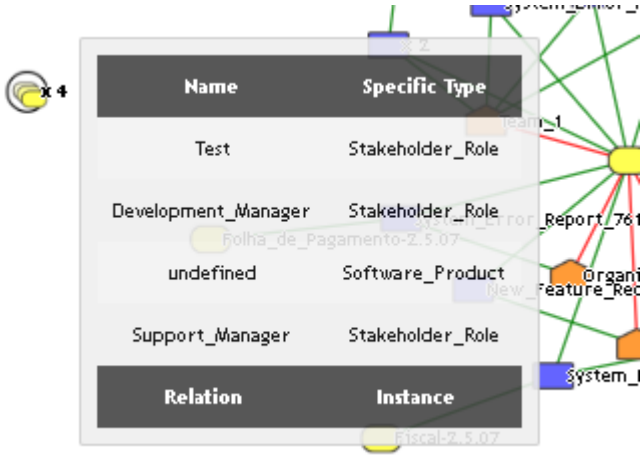


Figure H.1: SDP2 - Tooltip.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is correct, because in the provided data, the activities performed by the *Test*, *Development_Manager*, and *Support_Manager* were not informed (as well as artifacts manipulated by them). Regarding the nameless artifact, the manager assumed that this was also in the data, but he could not explain why it happened. Considering that this fact occurs with one artifact, the manager mentioned that this can be treated as an exception. It would be a concern only if such fact was recurring.

b) This analysis *can* assist in the proposed decision-making.

c) He *can partially* answer CQ1 using his current process management tool - he uses Mantis[31] as BugTracker and a tool developed in his company (called *Head*). Using *Head* he can identify all the elements involved in the process, however, he does not currently have a view (or graph) that relates all of them, as presented by the iSPuP approach.

d) Answering CQ1 is *extremely relevant* to support in analysis and decision-making processes.

– **CQ2** *Which procedures are used by the process during its execution?*

Using SDP2 provided data, CQ2 was not possible to be answered since no procedure was informed in the process execution data. However, as it was done with SDP1, we show to the process manager an analysis possibility in CQ2 using the toy

[31] https://www.mantisbt.org/

example, and the following answers were obtained from him (we do not make the question to check if the analysis is correct, considering he does not know in details the process used in the toy - only a quick explanation of it was provided at the beginning of the interview):

a) -
b) This analysis *can* assist in the proposed decision-making.
c) He *can partially* answer CQ2 through a manual analysis (using queries in SQL), since the company currently controls and stores the procedures used during the process execution (such fact did not occur when the execution data were provided for the analysis in this thesis).
d) Answering CQ2 is *extremely relevant* to support in analysis and decision-making processes.

– **CQ3:** *Which activities had a high complexity (considering the number of associated stakeholders, artifacts, procedures and / or resources)?*

In order to answer CQ3, the activity' degree was used. Figures H.2 and H.3 are generated by the visualization tool to support CQ3. The tabular view was used, and we filter all the activities. After that, we order the obtained results by its degree - firstly descending (Fig. 6.17) and, after that, ascending (Fig. 6.18). The first three results obtained are shown in these figures. According to what is shown, it was not possible to perceive any great discrepancy between the levels of activities analyzed (minimum 2 and maximum 3), therefore, it was concluded, through this analysis, that, according to this specific metric (activity grade), none of the activities performed during the 10 instances of the process can be considered more complex than the others.

id	Name	Type	Degree
0	Case_Resolution_7635	Activity	3
3	Case_Resolution_7631	Activity	3
7	New_Feature_Request_7609	Activity	3

Figure H.2: SDP2 – Activities Degree – Part 1.

id	Name	Type	Degree
2	Case_Registration_7602	Activity	2
10	Close_the_Case_7636	Activity	2
16	Close_the_Case_7632	Activity	2

Figure H.3: SDP2 – Activities Degree – Part 2.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*.
b) This analysis *can* assist in the proposed decision-making.
c) He *can* answer CQ3 using his current process management tool.
d) Answering CQ3 is *extremely relevant* to support in analysis and decision-making processes.

– **CQ4:** *Which activities had a high dependency (on other activities)?*

Activities dependency on other activities is provided by PROV-SwProcess Model using the relation *WasInformedBy* (implying that there has been the exchange of some artifact by two activities, one activity using (or changing) some artifact generated by the other activity) and is useful only when just one instance is analyzed. When checking SDP2 instances separately, no dependency between activities was found, because none of the artifacts generated during the 10 analyzed instances was used or modified by another activity in the same instance. We discussed this fact with the manager, and the following points should be considered:

a) This fact is *correct*. The manager mentioned that if test activity was considered, for example, this type of dependency would be shown (however, it should be noted that no data were provided regarding the test activity execution).
b) It *can* assist in the proposed decision-making.
c) He *can* answer CQ4 using his current process management tool.
d) Answering CQ4 is *extremely relevant* to support in analysis and decision-making processes.

Goal 2: Understanding stakeholder's involvement in process execution

– **CQ5:** *What is the activities distribution among stakeholders?*

Figure H.4 is generated by the visualization tool to support in answering CQ5.

id	Name	Type	Degree	Created Artifacts	Modified Artifacts	Associated Activities
22	Person_2	Person_Stakeholder	10	0	0	6
26	Person_5	Person_Stakeholder	13	3	1	4
71	Person_10	Person_Stakeholder	11	2	1	4
69	Team_1	Team_Stakeholder	9	0	0	4
23	Person_3	Person_Stakeholder	7	0	0	4
21	Person_1	Person_Stakeholder	8	0	1	3
13	Person_6	Person_Stakeholder	4	0	0	2
15	Person_7	Person_Stakeholder	4	0	0	2
16	Person_8	Person_Stakeholder	4	0	0	2
25	Person_4	Person_Stakeholder	4	0	0	2
42	Organization_3	Organization_Stakeholder	5	0	0	2
17	Person_9	Person_Stakeholder	4	0	1	1
39	Organization_1	Organization_Stakeholder	3	0	0	1
43	Organization_2	Organization_Stakeholder	3	0	0	1
45	Organization_5	Organization_Stakeholder	3	0	0	1
47	Organization_4	Organization_Stakeholder	3	0	0	1

Figure H.4: SDP2 - Stakeholders X Activities – Tabular View.

Considering the *Person* stakeholders, *Person_2* performed five activities more than *Person_9*, for example. Are these discrepancies really been planned or is it possible to have a better activity distribution among stakeholders during the process instantiation?

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*. The manager said that *Person_2* usually performs more activities because he is assigned only with activities with low level of difficulty, that can be quickly solved. *Person_9*, however, is a system expert and often receives activities that are more difficult and require more time. He mentioned that this analysis would be more complete considering the time spent on each activity.

b) This analysis *can* assist in the proposed decision-making.

c) He *cannot* answer CQ5 using his current process management tool.

d) Answering CQ5 is *somewhat relevant* to support in analysis and decision-making processes. When he gave this response, the manager mentioned that only if the estimated effort and the time spent on each activity was considered, answering CQ5 would be extremely relevant to support in the proposed analysis and decision-making.

– **CQ6:** *Which artifacts are known by a stakeholder, considering that in some process execution he/she created or modified such artifact?*

Figure H.5 is an example of visualization generated by the tool to support answering CQ6. According to this figure, we can see, for example, that only *Person_5* stakeholder manipulated some artifacts like *Honorarios-2.5.08* and *Geral-2.5.08* during the ten analyzed instances, then, we may consider that *Person_5* have some knowledge about them. In the other hand, the artifact *Fiscal-2.5.09* was manipulated both by *Person_5* and *Person_1*, for example. Several other analyzes can be made from the Figure H.5, in relation to artifacts known to stakeholders. In a future instantiation of the analyzed process, if a certain task is associated with a specific artifact (*Fiscal-2.5.09* for example), the process manager can allocate this task to *Person_5* or *Person_1*, considering both of them know this artifact.

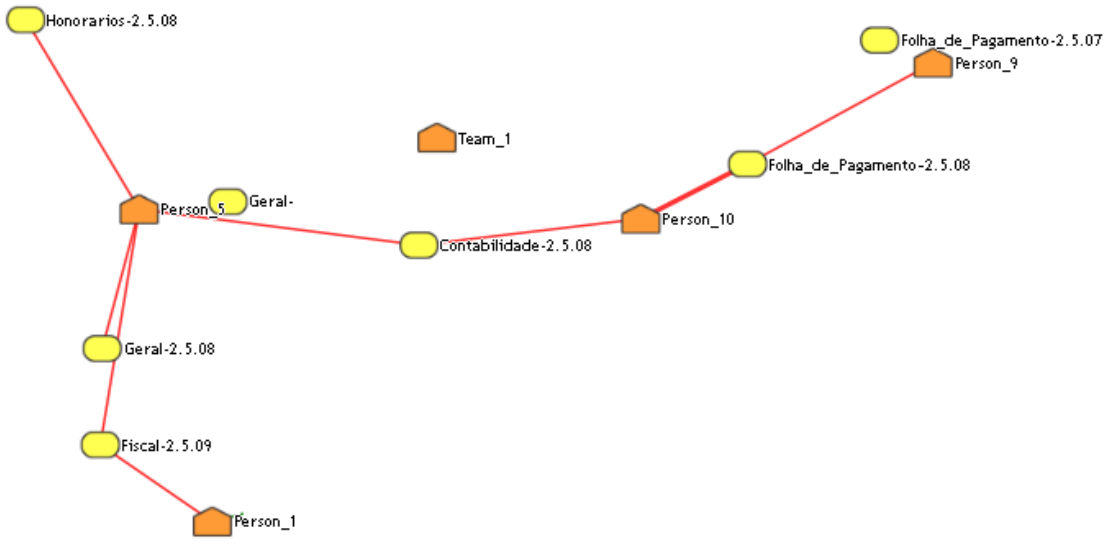


Figure H.5: SDP2 – All Stakeholders X Artifacts.

When this analysis was presented to the manager, the following answers were obtained:

a) The presented analysis is *correct*.

b) This analysis *can* assist in the proposed decision-making.

c) He *cannot* answer CQ6 using his current process management tool.

d) Answering CQ6 is *very relevant* to support in analysis and decision-making processes.

– **CQ7:** *Which stakeholders are out of the average of created and/or modified artifacts?*

Figure H.6 is generated by the visualization tool to support in answering CQ7. *Person_5* created 3 artifacts and modified 1 and *Person_10* created 2 artifacts and modified 1. All other stakeholders have only modified one or no artifact. Considering these numbers, we cannot perceive any great discrepancy between the number of artifacts created or modified by the stakeholders and no decision-making is suggested. It is believed (based on the other two case studies scenarios) that if more instances were analyzed, this discrepancy could occur.

id	Name	Type	Degree	Created Artifacts	Modified Artifacts
26	Person_5	Person_Stakeholder	13	3	1
71	Person_10	Person_Stakeholder	11	2	1
69	Team_1	Team_Stakeholder	9	0	0
13	Person_6	Person_Stakeholder	4	0	0
15	Person_7	Person_Stakeholder	4	0	0
16	Person_8	Person_Stakeholder	4	0	0
17	Person_9	Person_Stakeholder	4	0	1
21	Person_1	Person_Stakeholder	8	0	1
22	Person_2	Person_Stakeholder	10	0	0
23	Person_3	Person_Stakeholder	7	0	0
25	Person_4	Person_Stakeholder	4	0	0
39	Organization_1	Organization_Stakeholder	3	0	0
42	Organization_3	Organization_Stakeholder	5	0	0
43	Organization_2	Organization_Stakeholder	3	0	0
45	Organization_5	Organization_Stakeholder	3	0	0
47	Organization_4	Organization_Stakeholder	3	0	0

Figure H.6: SDP2 - Stakeholders X Created and Artifacts – Tabular View.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*.
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ7 using his current process management tool.
d) Answering CQ7 is *very relevant* to support in analysis and decision-making processes.

– **CQ8:** *What are the relationships among stakeholders?*

Using SDP2 provided data, CQ8 was not possible to be answered since no relation among stakeholders' roles was informed in the process execution data and this information was not inferred by PROV-SwProcess. However, we showed to the process manager an analysis possibility in CQ8 using the toy example, and the following answers were obtained from him:

a) -
b) This analysis *can partially* assist in the proposed decision-making. According to the manager, it is not enough to know the relationship between the stakeholders, it would be necessary to understand the level of this relationship and the reason why it occurs.
c) He *cannot* answer CQ8 using his current process management tool
d) Answering CQ8 is *somewhat relevant* to support in analysis and decision-making processes, considering what he said in b).

– **CQ9:** *Which roles does each stakeholder assume?*

Figure H.7 is generated to support in answering CQ9. Using this figure, we can see all the stakeholders that acts as a *Programmer*, as *Support* or as a *Client*. The group of roles in the lower corner of the figure corresponds to the three roles informed in the process model which had no associated stakeholder (based on the provided execution data). According to this figure, we can also see that the most versatile stakeholder is *Person_1*, who acts as *Programmer* and as *Support*, according to the process provenance data. In a next instantiation of this process, if the process manager needs to allocate a *Programmer* or a *Support* person in a specific activity, he knows who can perform these roles, based on previous execution data.

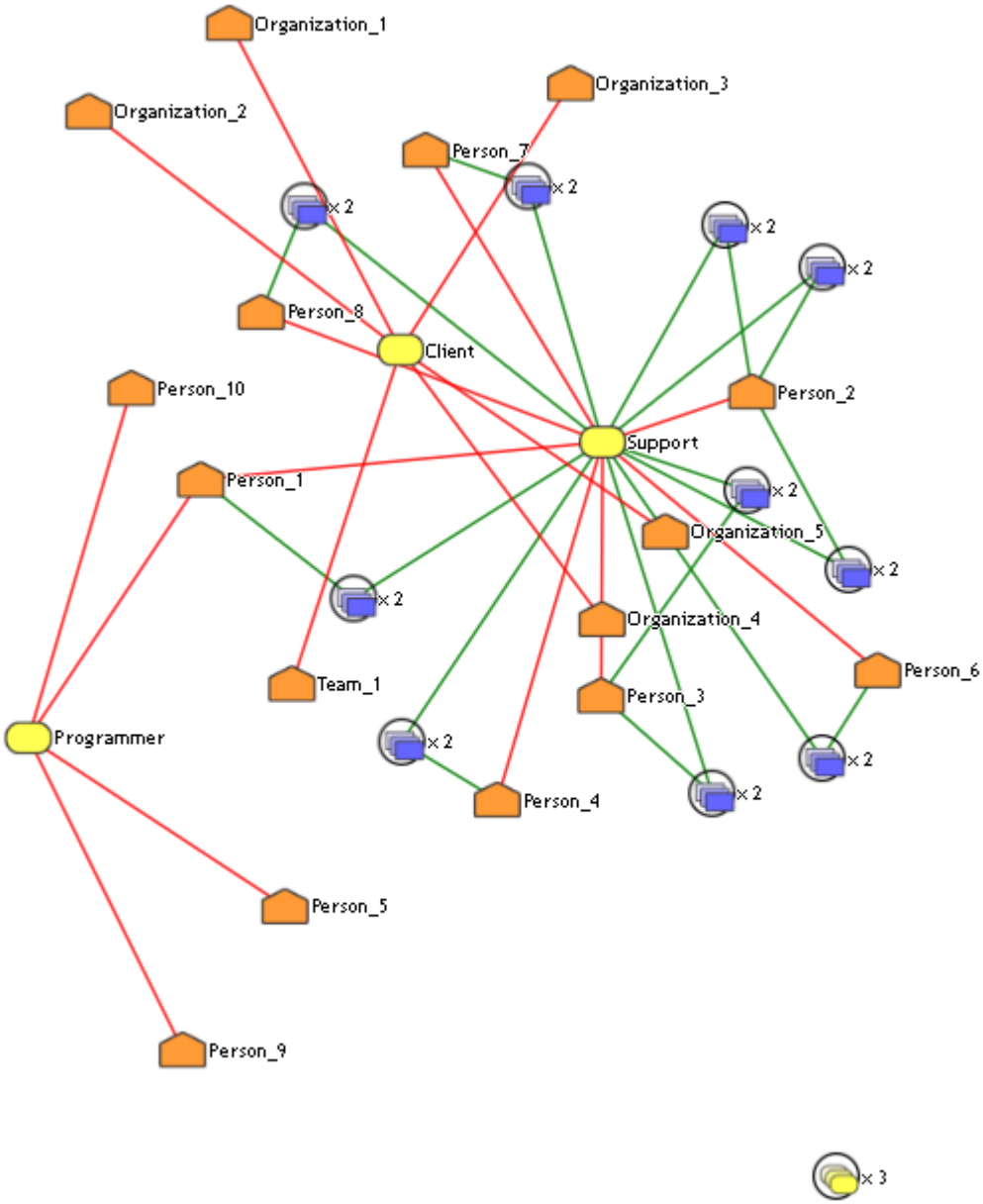


Figure H.7: SDP2 - Stakeholders x Roles.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*; however, it is not common during the process execution that a stakeholder assumes both a *Support* and a *Programmer* role. Besides that, he agreed that this occurred in one of the analyzed instances.

b) This analysis *can* assist in the proposed decision-making.

c) He *cannot* answer CQ9 using his current process management tool.

d) Answering CQ9 is *somewhat relevant* to support in analysis and decision-making processes.

Goal 3: Tracking derivations and revisions among artifacts or procedures

– **CQ10:** *Which artifacts are derivations from others?* and

– **CQ11:** *Which artifacts or procedures are revisions from others?*

Figure H.8 is generated by the visualization tool to support the achievement of GOAL 3 (CQ10 and CQ11). Considering this visualization, no derivation between artifacts was found (no association was inferred among the artifacts manipulated by the 10 instances of this process). Although the artifacts used, created and modified by the activities have been informed, it is believed that due to the small number of instances analyzed (10), this fact did not occur.

When this analysis was presented to the manager, the following answers were obtained (both for CQ10 and CQ11):

a) This analysis is *correct*, considering only the analyzed group of data.
b) This analysis *can* assist in the proposed decision-making.
c) He *can* answer CQ10 and CQ11 using his current process management tool.
d) Answering CQ10 and CQ11 is *very relevant* to support in analysis and decision-making processes.

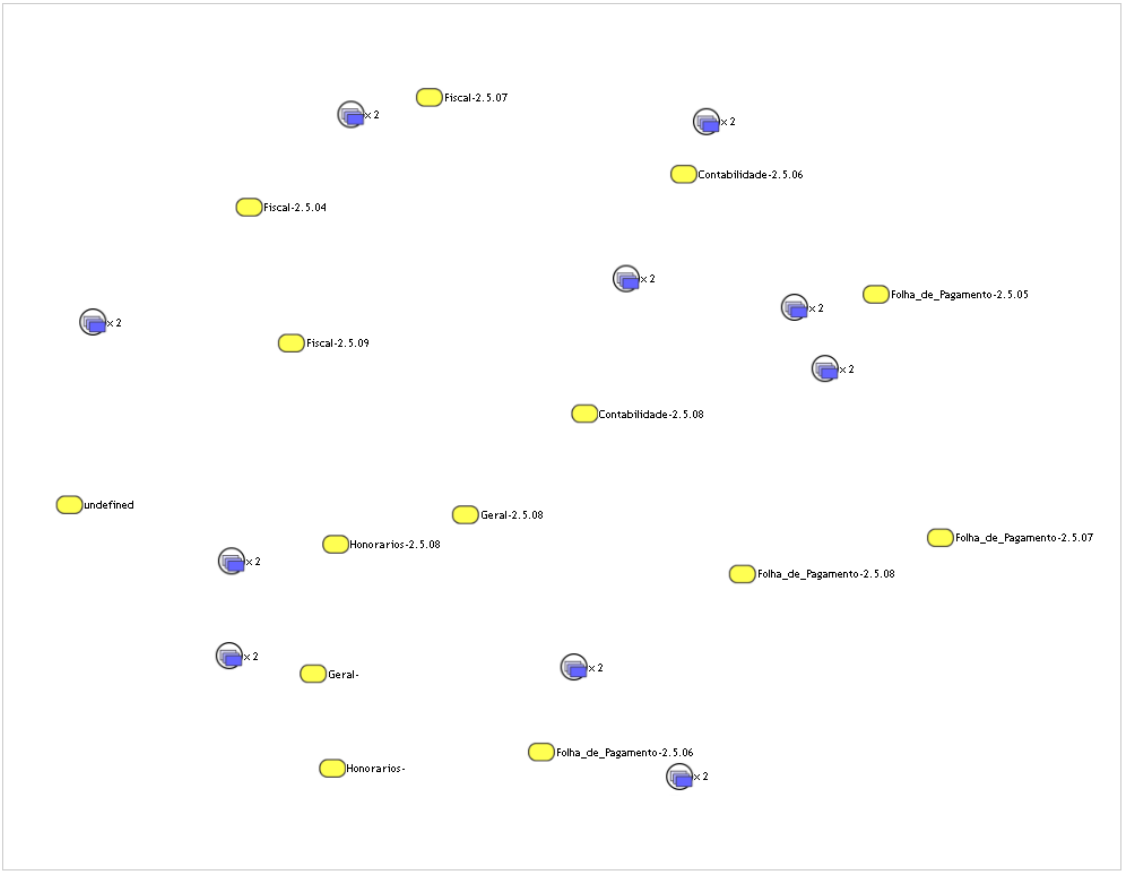


Figure H.8: SDP2 – Artifacts Derivation.

APPENDIX I - SDP3: DETAILED EXECUTION AND ANALYSIS

This Appendix presents a detailed discussion about all the PROV-SwProcess Competency Questions, using data from SDP3 and including the manager's opinion on each of them (the same procedure reported for SDP1 was followed, changing only the scenario and the subject).

Goal 1: Process structure identification during execution and possibilities for process redesign

- **CQ1** *What are the process activities, artifacts, resources, procedures, stakeholders, and the relations among them during the process execution?*

Considering the analysis possibility presented by CQ1 (*It is possible to identify all the process elements that participated in the process execution and the relation among them*) and its respective decision-making possibility (*After identifying the process elements and the relation between them it is possible to find gaps – elements without association or inadequate relation established*), when using iSPuP tool support and the generated macro visualization about SDP3 (shown on Figure 7.15), a stakeholder was identified as 'NULL' (an orange pentagon in the lower center of the figure), associated with some tasks and artifacts. When hovering the mouse over it, it was verified that this stakeholder acted as a developer, modifying versions 0, 3 and 5 of the analyzed project (Figure I.1). Based on this, we consider this fact as a gap possibility: some unidentified person performed these tasks and changed project artifacts and this information was not properly stored. If this is recurring, this can lead to problems in the project development, such as tracking people who have modified a particular artifact or performed some activity.

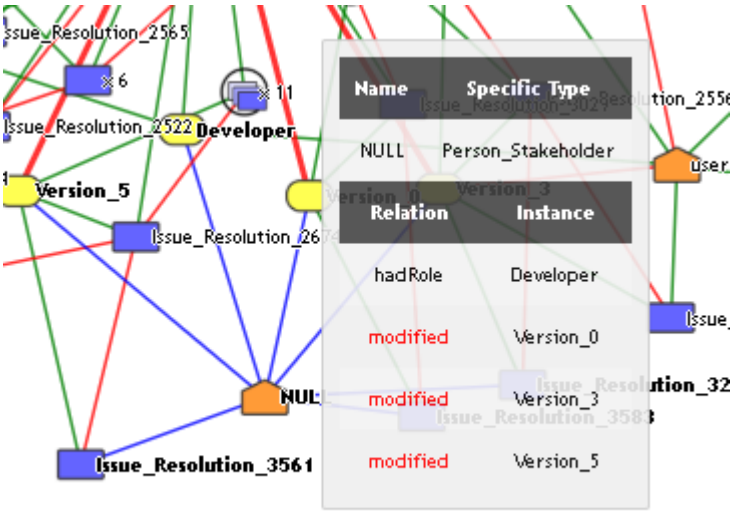


Figure I.1: SDP3 – Null Stakeholder.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*.
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ1 using his current process management tool.
d) Answering CQ1 is *extremely relevant* to support in analysis and decision-making processes.

– **CQ2** *Which procedures are used by the process during its execution?*

Using SDP3 provided data, CQ3 was not possible to be answered since no procedure was informed in the process execution data. However, as it was done with SDP1, we showed to the process manager an analysis possibility in CQ2 using the toy example, and the following answers were obtained from him (we do not make the question to check if the analysis is correct, considering he does not know in details the process used in the toy - only a quick explanation of it was provided at the beginning of the interview):

a) -
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ2 using his current process management tool.
d) Answering CQ2 is *very relevant* to support in analysis and decision-making processes.

— **CQ3:** *Which activities had a high complexity (considering the number of associated stakeholders, artifacts, procedures and / or resources)?*

In order to answer CQ3, the activity' degree was used. Figures I.2 and I.3 are generated by the visualization tool to support CQ3. The tabular view was used, and we filter all the activities. After that, we ordered the obtained results by its degree - firstly descending (Fig. I.2) and, after that, ascending (Fig. I.3). According to what is shown, it is possible to see a great discrepancy between the degree of *Issue Resolution* activities and the other two (*Issue Registration* and *Issue Attribution*).

id	Name	Type	Degree
65	Issue_Resolution_2556	Activity	17
72	Issue_Resolution_2674	Activity	16
111	Issue_Resolution_2522	Activity	16
151	Issue_Resolution_2853	Activity	16
107	Issue_Resolution_2526	Activity	13
87	Issue_Resolution_2543	Activity	11
105	Issue_Resolution_2525	Activity	9
264	Issue_Resolution_3083	Activity	8
46	Issue_Resolution_2565	Activity	6
161	Issue_Resolution_2727	Activity	6

Figure I.2: SDP3 – Activities Degree – Part 1.

id	Name	Type	Degree
1	Issue_Registration_2718	Activity	1
5	Issue_Attribution_3631	Activity	1
6	Issue_Attribution_3632	Activity	1
8	Issue_Attribution_2543	Activity	1
12	Issue_Attribution_2536	Activity	1
13	Issue_Attribution_3624	Activity	1
14	Issue_Registration_2722	Activity	1
15	Issue_Registration_2724	Activity	1
17	Issue_Registration_2727	Activity	1
19	Issue_Attribution_3640	Activity	1

Figure I.3: SDP3 – Activities Degree – Part 2.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *partially correct*. He justified his answer saying that only the degree of the activity cannot be determinant to evaluate its complexity. This may be an 'indicator' of that, but, in order to assert this, it would require a more in-depth analysis considering the activity to be developed.
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ3 using his current process management tool.
d) Answering CQ3 is *very relevant* to support in analysis and decision-making processes.

— **CQ4:** *Which activities had a high dependency (on other activities)?*

Activities dependency on other activities is provided by PROV-SwProcess Model using the relation *WasInformedBy* (implying that there has been the exchange of some artifact by two activities, one activity using (or changing) some artifact generated by the other activity) and is useful only when just one instance is analyzed. When checking SDP3 instances separately, no dependency between activities was found, because none of the artifacts generated during the 133 analyzed instances was used or

modified by another activity in the same instance. We discussed this fact with the manager, and the following points should be considered:

a) This analysis is *correct*.
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ4 using his current process management tool.
d) Answering CQ4 is *extremely relevant* to support in analysis and decision-making processes.

Goal 2: Understanding stakeholder's involvement in process execution

– **CQ5:** *What is the activities distribution among stakeholders?*

Figure I.4 is generated by the visualization tool to support in answering CQ5.

id	Name	Type	Degree	Created Artifacts	Modified Artifacts	Associated Activities
42	user_29	Person_Stakeholder	391	18	16	368
43	user_26	Person_Stakeholder	19	1	1	14
357	user_41	Person_Stakeholder	24	1	8	14
248	NULL	Person_Stakeholder	7	0	3	3

Figure I.4: SDP3 - Stakeholders X Activities – Tabular View.

Considering the *Person* stakeholders, while *user_26* and *user_41* performed the same number of activities, *user_29* executed 354 more activities than them. Through this tabular visualization, a poor activity distribution between these three stakeholders is clearly perceived. The visualization of which these activities are can be obtained using the graph representation (filtering only the stakeholders and the activities).

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*.
b) This analysis *can partially* assist in the proposed decision-making. Considering the presented decision-making possibility, it should be considered not only the number of associated activities, but also the time of accomplishment of these activities.
c) He *cannot* answer CQ5 using his current process management tool.
d) Answering CQ5 is *extremely relevant* to support in analysis and decision-making processes.

– **CQ6:** *Which artifacts are known by a stakeholder, considering that in some process execution he/she created or modified such artifact?*

Figure I.5 is an example of visualization generated by the tool to support answering CQ6. According to this figure, we can see, for example, that both *user_26* and *user_29* manipulated *Version_1* and *Version_3* and only stakeholder *user_26* manipulated *Version_2*, *Version_15*, *Version_16*, and *Version_21*, for example. Specifically, in the case of SDP3, no data were provided of which project-specific artifacts were created, modified or changed. Only the information of the project versions that were created, altered or used during the activities was provided for this analysis.

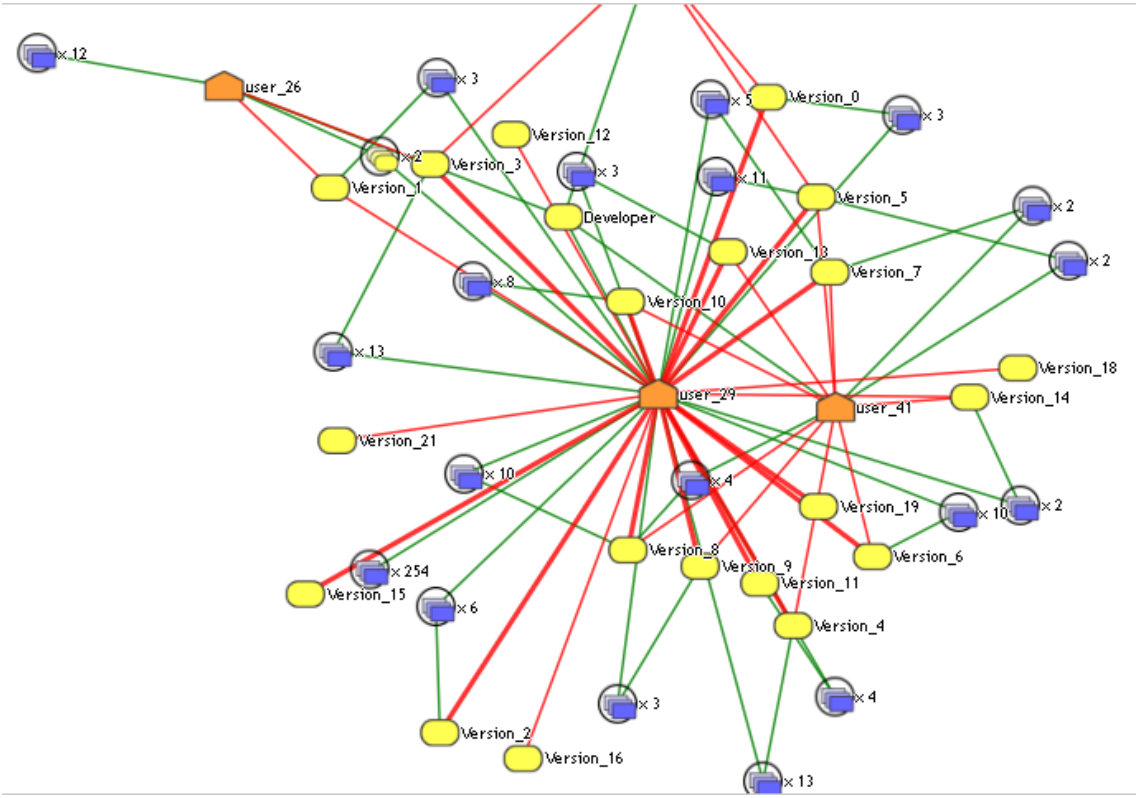


Figure I.5: SDP3 – Stakeholders and Associated Artifacts.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*.

b) This analysis *can partially* assist in the proposed decision-making.

c) He *cannot* answer CQ6 using his current process management tool.

d) Answering CQ6 is *extremely relevant* to support in analysis and decision-making processes.

— **CQ7:** *Which stakeholders are out of the average of created and/or modified artifacts?*

In the case of SDP3, the same visualization generated for CQ6 (Figure I.5) can be used for CQ7 and we can easily see that the stakeholder *user_26* stands out from the other stakeholders when the number of manipulated artifacts is analyzed. The exact number of manipulated artifacts can be obtained through the tabular view (Figure I.6). As a decision-making possibility in relation to this fact, it is suggested a better distribution of the activities among the stakeholders because, considering that much more activities were assigned to this user (*user_26*), this made him/her manipulate much more artifacts than the other users.

id	Name	Type	Degree	Created Artifacts	Modified Artifacts
42	user_29	Person_Stakeholder	391	18	16
43	user_26	Person_Stakeholder	19	1	1
357	user_41	Person_Stakeholder	24	1	8
248	NULL	Person_Stakeholder	7	0	3

Figure I.6: SDP2 - Stakeholders X Created and Artifacts – Tabular View.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*.
b) This analysis *can* assist in the proposed decision-making.
c) He *cannot* answer CQ7 using his current process management tool.
d) Answering CQ7 is *not very relevant* to support in analysis and decision-making processes.

— **CQ8:** *What are the relationships among stakeholders?*

Using SDP3 provided data, CQ8 was not possible to be answered since no relation among stakeholders' roles was informed in the process execution data and this information was not inferred by PROV-SwProcess. However, we showed to the process manager an analysis possibility in CQ8 using the toy example, and the following answers were obtained from him:

a) -
b) This analysis *can* assist in the proposed decision-making.

c) He *cannot* answer CQ8 using his current process management tool.

d) Answering CQ8 is *extremely relevant* to support in analysis and decision-making processes.

— **CQ9:** *Which roles does each stakeholder assume?*

Figure I.7 is generated to support in answering CQ9. Using this figure, we can see that *user_26* and *user_29* performed all the 3 process roles, while *user_41* and the 'NULL' user (presented in the CQ1 analysis) acted only as developer. As a decision-making possibility the manager can check if this fact was really planned and, if yes, in a next instantiation of this process, if the process manager needs to allocate a *Programmer, a Manager or a Developer* in a specific activity, he knows who can perform these roles, based on previous execution data.

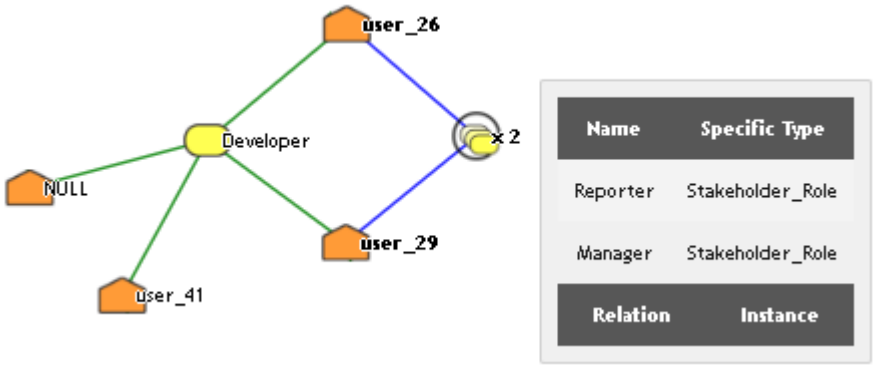


Figure I.7: SDP2 - Stakeholders x Roles.

When this analysis was presented to the manager, the following answers were obtained:

a) This analysis is *correct*.

b) This analysis *can* assist in the proposed decision-making.

c) He *cannot* answer CQ9 using his current process management tool.

d) Answering CQ9 is *very relevant* to support in analysis and decision-making processes.

Goal 3: Tracking derivations and revisions among artifacts or procedures

— **CQ10:** *Which artifacts are derivations from others?* and

— **CQ11:** *Which artifacts or procedures are revisions from others?*

Considering SDP3, no derivation between artifacts was inferred. This fact occurs because only the information about created and changed versions was informed, and no information was provided about what versions were used by the activities.

When this analysis was presented to the manager, the following answers were obtained (both for CQ10 and CQ11):

a) This analysis is *correct*, considering only the analyzed group of data.
b) This analysis *can* assist in the proposed decision-making.
c) He *can* answer CQ10 and CQ11 using his current process management tool.
d) Answering CQ10 and CQ11 is *extremely relevant* to support in analysis and decision-making processes.