2016

Continuous Process Auditing (CPA): an Audit Rule Ontology Approach to Compliance and Operational Audits

Numanul Hoque Subhani
University of Windsor

Continuous Process Auditing (CPA): an Audit Rule Ontology Approach to Compliance and Operational Audits

by

K M Numanul Hoque Subhani

A Dissertation
Submitted to the Faculty of Graduate Studies
through the **School of Computer Science**
in Partial Fulfillment of the Requirements for
the Degree of **Doctor of Philosophy** at the
University of Windsor

Windsor, Ontario, Canada
2016

Continuous Process Auditing (CPA): an Audit Rule Ontology Approach to Compliance and Operational Audits

by

K M Numanul Hoque Subhani

APPROVED BY:

Dr. Kumaraswamy Ponnambalam, External Examiner
Department of Systems Design Engineering, University of Waterloo, Ontario, Canada

---

Dr. Eksa Kilfoyle, External Reader
Department of Accounting, Odette School of Business

Dr. Scott Goodwin, Internal Reader
School of Computer Science

Dr. Jianguo Lu, Internal Reader
School of Computer Science

Dr. Robert Kent, Advisor
School of Computer Science

May 16^{th}, 2016

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

This thesis also incorporates the outcome of a joint research under the supervision of professor Dr Robert Kent. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of co-authors was primarily through the provision of guidance corrections and constructive criticism.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publications

This thesis includes two original papers that have been previously published/submitted for publication in peer reviewed journals (see next page - Table I):

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, tech-

Table 1: Declaration of Previous Publications

Thesis Chapter	Publication title/full citation	Publication status
Chapters 6, 7	***N. Subhani, R. Kent**: Novel design approach to build audit rule ontology for healthcare decision support systems. Intl Conf. e-Learning, e-Bus., EIS, and e-Gov. (EEE14). 133–138, 2014*	*Published*
Chapters 6, 7	***N. Subhani, R. Kent**: Incorporating policy-based authorization framework in audit rule ontology for continuous process auditing in complex distributed systems. Intl Workshop on Info. Sys. in Distributed Environment (ISDE14), LNCS 8842:367–376, 2014*	*Published*
Chapters 10	***N. Subhani, R. Kent**: Continuous Process Auditing (CPA): an audit rule ontology approach to Audit-as-a-Service. IEEE International Systems Conference (SysCon15):832–838, 2015*	*Published*

niques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Continuous Auditing (CA) has been investigated over time and it is, somewhat, in practice within financial and transactional auditing as a part of continuous assurance and monitoring. Enterprise Information Systems (EIS) that run their activities in the form of processes require continuous auditing of a process that invokes the action(s) specified in the policies and rules in a continuous manner and/or sometimes in real-time. This leads to the question: How much could continuous auditing mimic the actual auditing procedures performed by auditing professionals? We investigate some of these questions through Continuous Process Auditing (CPA) relying on heterogeneous activities of processes in the EIS, as well as detecting exceptions and evidence in current and historic databases to provide audit assurance.

In this dissertation research, we propose a shared vocabulary, which defines the processes, and the audit rule ontology for process (AROP) that would be used to detect exceptions in a process. Both of these are integrated to form a hybrid ontology that supports the audit rule acquisition and the evolution of ontologies. We also devised a semi-automatic mechanism to construct a common ontology by coupling to an expert system. These CPA methodologies are subject to experimental testing in three different pervasive environments, including contin-

uous assurance and monitoring of healthcare decision support, e-commerce, and production system processes. A direction of various applications of audit rules, AROP, and audit rule based systems is also presented.

Dedication

To my mother whom I loved most,

for her unconditional love,

support and years of expectation ...

To my departed father,

for immortal inspirations

and endless encouragements ...

Acknowledgements

Firstly, I would like to express my deepest appreciation and gratitude to my advisor Dr. Robert D. Kent for his encouragement, support and invaluable suggestions in guiding me towards the successful completion of this research work. Without his generous funding and advice, it would have been hard for me to achieve publications from this work. His guidance has constantly encouraged me in both personal and academic life to build my personality as a young research professional.

I would like to express my gratitude to all members of my doctoral committee: Dr. Kumaraswamy Ponnambalam, Dr. Scott Goodwin, Dr. Jianguo Lu, and Dr. Eksa Kilfoyle. They sacrificed a lot of valuable time in reviewing my dissertation, giving me constructive feedback, and attending my seminars, comprehensive examination, proposal, and defense.

I also acknowledge scholarships I have received from Ontario Graduate Scholarship in Science and Technology (2010-2011), Ontario Graduate Scholarships (2011-2012, 2014-2015), Postgraduate Scholarship - Doctoral (PGSD) (2012-2014) from The Natural Sciences and Engineering Research Council (NSERC) of Canada, and tuition scholarships (2010-2013) from The University of Windsor and conference travel grants for Enterprise Information Systems (EEE'14), IEEE

System Conference (SysCon'15) conferences.

I would like to express my gratitude for the financial assistance I have received through my supervisor's research grants. In particular, I acknowledge ongoing support from the Canadian Institutes for Health Research (CIHR) team grant in Traffic and Road Injury Prevention (TRIP) (2011-2016), with PI's Dr. Anne W. Snowdon (UWindsor) and Dr. Andrew Howard (SickKids Hospital, Toronto), and the Cross Border Institute (CBI) at the University of Windsor (2013-2014), with PI Dr. William Anderson (UWindsor).

I would like to thank my parents. Without their encouragement, support and love, it would not have been possible for me to pursue so many great achievements in my life.

Many thanks to *Amar Bou* for her support, co-operation, sacrifice and being patient with me ... ... :)

Finally, I want to extend my gratitude to my friends, the faculty members and staff of the School of Computer Science for their friendly suggestions and support during my study at the University of Windsor.

Contents

List of Figures

List of Tables

Acronyms

AICPA American Institute of Certified Public Accountants. 18, 42

AROP Audit Rule Ontology of a Process. 7, 69, 70, 146, 147

BCM Business Continuity Management. 52

CA Continuous Auditing. 8, 18, 31, 145, 146, 151

CAAT Computer Assisted Audit Tools. 21

CAWS Continuous Auditing Web Service. 41

CCM Continuous Control Monitoring. 6, 8, 31, 132, 133

CDA Continuous Data Assurance. 6, 31

CEP Complex Event Processing. 35

CICA Canadian Institute of Chartered Accountants. 18, 42

CM Continuous Monitoring. 33, 145, 151

CO Common Ontology. 61, 63, 146, 150

CPA Continuous Process Auditing. 3, 5, 6, 8, 9, 30, 59, 62, 68, 69, 85, 90, 132, 133, 136, 138, 140, 141, 143, 145–151

CPAS Continuous Process Auditing System. 20, 27

DAG Directed Acyclic Graph. 62

DFD Dataflow Diagram. 62

DL Descriptive Logic. 137, 139, 148, 155

EAM Embedded Audit Module. 20

ECA Event-Condition-Action. 35

ECAP Event-Condition-Action-Postcondition. 35

EIS Enterprise Information System. 78

ERP enterprise Resource Planning. 78

GAAP Generally Accepted Accounting Principles. 11, 17

GAAS Generally Accepted Auditing Standards. 50

GTAG Global Technology Audit Guide. 18

ISACA Information Systems Audit and Control Association. 19

KR Knowledge Representation. 35

MO Mapping Ontology. 59, 74

MTSM Multivariate Time Series Model. 45

NLP Natural Language Processing. 139

OWL Wen Ontology Language. 79, 81

PAA Predictive Auditing Analytics. 143, 149, 150

PO Process Ontology. 63, 74, 98–100, 146

RDFS Resource Description Framework Schema. 79, 81

RIF Rule Interchange Format. 80

RuleML The Rule Markup Language. 126, 154, 155

SAS Statement on Auditing Standards. 43

SEM Simultaneous Equation Model. 45

SOA Service-Oriented Architecture. 16

SOX Sarbanes-Oxley Act. 21

URI Uniform Resource Identifier. 81

XBRL eXtensible Business Reporting Language. 22

XCAL eXtensible Continuous Auditing Language. 40

Chapter 1

Introduction

Over the last few decades, Information Technologies have provided an important means for business process interchange and integration. To assure better corporate and system accountability, traditional periodic auditing that adopts a backward-looking approach, whereby key activities and events are often identified long after their occurrence or simply undetected, has been implemented for the most part. Continuing changes and innovations in corporate culture, and recent developments and advances in information technology in conjunction with real-time approaches to conducting business, are challenging the auditing profession.

In straightforward terms, auditing encompasses a variety of methods used to measure the compliance of a system to defined rules and guidelines. Auditing is just one facet of a more extensive set of processes, often based on accounting, but extending to support assurance that the system is working as intended. Clearly, auditing is applied in many domains, from government to business, education and health care, among many others. Underscoring

the breadth of auditing applications is the fact that most auditing is still performed by human agents, professionally trained and experienced in many aspects of evidence gathering, interpretation of rules and guidelines, and clarifying of final reports in respect of limitations.

Increasingly, complex systems have grown beyond the capacities of human driven auditing to perform meaningful audits in a timely fashion that serves stakeholders and oversight bodies. Such systems encompass networks of human agents and also highly automated software systems with semi-autonomous sub-systems, all of which are assumed to be imperfect or vulnerable to risk. To this end, researchers have focused attention on the notion of automating significant parts of auditing, both as embedded components within systems working autonomously, and as decision support components serving human analysts.

One vital element throughout auditing concerns knowledge, namely, its acquisition, interpretation and uncertainty, or vagueness, in reasoning. From the outset, auditing practice dictates that meaningful definitions must be determined and documented, for the system to be audited, components and processes within the system, evaluation measures, rules and actions to be applied, and limitations or constraints. In all aspects, auditors must work with suitable knowledge expressed in natural language terms for human consumption, but also expressed in terms appropriate for application and reasoning through logic, using computational techniques in particular.

Knowledge is an essential part of most Semantic Web applications and ontology, which is a formal explicit description of concepts or classes in a domain of discourse [36], is an essential part of the knowledge. Extracting

knowledge from text in a semi-automatic way and identifying effective procedures for achieving useful and reliable results are challenging and daunting scientific research areas. In the auditing field, most audit rules are defined in the context of human understanding and language and can be used to support human cognitive reasoning. Inference based on interpretations of those audit rules are as essential to the Semantic Web as application domain ontologies. Ontology-based reasoning has known shortcomings and limitations compared with rule-based reasoning [38]. To represent inferential knowledge, ontology alone is insufficient [37]; but, inferential rules are an essential part of the knowledge in an audit rule ontology for a process or module in Continuous Process Auditing (CPA) for real-time Decision Support systems [13].

Berners-Lee [14] defined the semantic web as an extension of the current content-based web, in which information is given well-defined meaning. An ontology is usually defined as a formal specification of domain knowledge conceptualization. Ontology learning by application of semi-automatic methods has been studied and most techniques are from free natural language text. Though chronological, topological, and other types of semantic relations already exist [74], in these methods only hierarchical concepts are extracted and reduced sets of semantic relations are in use.

In Continuous Process Auditing methodology, an audit rule sheet is defined for each process or module. The matter of continuity of application in CPA ranges from continuous time-dependent modeling to discrete time steps of audit application adapted to application requirements. For approaches with small time steps, the sheer magnitude of computational tasks to support CPA demands use of coarse-grained analysis of sub-systems, and estimation tech-

niques based on limited rule sets. This consideration is used to determine the degree of conceptualization as knowledge, audit measures using sensors and reasoning through rules and inference.

1.1 Research Questions and Contributions

In the following, we will give an overview of our work in terms of research challenges and contributions. Section 1.1.1 discusses the most important questions that we will try to answer in this thesis. A summary of our contributions with respect to these research questions is listed in Section 1.1.2.

1.1.1 Research Questions

I - Heterogeneous Accessibility: Information systems are complex and heterogeneous in nature and their data sources are semantically heterogeneous as well. Continuous Process Auditing requires accessing to heterogeneous data sources in various and/or multiple locations.

(a) A common ontology approach is straightforward for dealing with heterogeneous semantic data sources. Hybrid approaches and multiple ontologies to the heterogeneity problem of ontologies have been discussed [85, 46]. Recently, we have proposed a hybrid audit rule ontology approach that coupled with an expert system to infer new relations from the existing concepts [71].

(b) In any autonomous pervasive distributed systems (like hospitals, clinics, nuclear power plants, any time-relay industrial/manufacturing set-

tings), accessing authorized data in real-time and enforcing data security as highly encrypted-sensitive data are transported from one layer to another in various (may be different geographic) locations are very important and obvious issues to handle before CPA is deployed for actions in audit rule ontology.

II - Semi-Automatic Hybrid Ontology Development: Many reasoning based applications demand for quality standards that are unattainable by automated ontology development methods. Developing ontology from multiple data sources through complex axioms in a fully automatic manner is often not feasible. In addition, certain level of expert knowledge is required through out the refinement and validation of ontology development process.

(a) We present a semi-automatic hybrid ontology development approach that combines the common ontology (shared vocabulary) with multiple, integrated and mapped Process Ontologies. This common ontology would be the top-most layer in topological semantic relations.

(b) Constant changes of business activities requires the change of business rules as well as audit rules for Continuous Process Auditing (CPA). The evolutionary process, a crucial part of the ontology lifecycle, creates newer versions with added stemming down the tree from the original ontology. Since the uniformity and coherence of the ontology must be respected, the evolution process is difficult to implement semi-automatically and should be considered as beyond human capacities for complex ontologies. Audit

Rule Ontologies for Processes (AROPs) would be stemmed as the second layer, under the common ontology layer in CPA, to infer the audit rules.

III - Audit Rules and Audit Assurance: As evolving business rules develop for incremental business needs, audit rules are usually written in abstract and human readable format. These audit rules are transformed into RuleML form which need to be inferred by rules engine for the part of audit assurance process. Inferring the audit rules and taking action accordingly are the two significant steps towards Continuous Control Monitoring (CCM) and Continuous Data Assurance (CDA). We investigated the uses of rule engines to process audit rules for CCM and we also devised a methodology for providing CDA.

1.1.2 Contributions

So far, Continuous Process Auditing (CPA) has been limited to the auditing professional communities and/or corporate world. In some cases, those approaches have barely scratched the surface of CPA potential, especially, in autonomous pervasive distributed systems that deal with heterogeneous data sources. This thesis introduces a benchmark to augment auditing professionals in Continuous Process Auditing by presenting methods and tools for

- accessing distributed data sources in way that eliminates the hurdle of data heterogeneity, i.e. semantically by hybrid ontology of domain and processes **(Ia,Ib)**,

- developing hybrid layered ontology semi-automatically that coupled with expert system for refinement **(IIa)**, and
- developing rule-based approach that ensures the continuous auditing of a process seamlessly, i.e. Audit Rule Ontology of Process (AROP) **(IIb)**.

Finally, this thesis also investigates the uses of rule engine to process audit rules for CCM and the uses of predicting methodologies for CDA to provide audit assurance **(III)**. In conclusion, we can state that: a CPA system is one that detects and obtains evidence associated with a process for the purpose of augmenting and assisting audit professionals in operational and compliance auditing in any EIS, and which invokes actions specified by and derived from policies and rules in a continuous manner. This dissertation research has produced a coherent framework design with implemented software modules that clearly demonstrate our successful outcomes.

1.2 Readers Guide

In the section, we give a brief overview of this dissertation. We organized the whole dissertation into four content Parts and appendices, and in the following we provide the reader with some basic orientation to our organization.

Background and Foundations (Part I):

In the first part of this dissertation, we lay the conceptual foundations for what follows, focusing on emerging concepts and methodologies for Continuous Auditing and Continuous Control Monitoring. After a general overview

and background review of CA in Chapter 2, we give a taxonomic analysis of Continuous Auditing and how the Continuous Process Auditing (CPA) has emerged in the research communities in Chapter 3. In Chapter 4, we provide a comparative analysis of CPA and the "Necessity of CPA in Compliance and Operational Audit". We review the relevant approaches and existing technologies to the Continuous Auditing (CA), Continuous Control Monitoring (CCM), and Continuous Process Auditing (CPA) in Chapter 5. Finally, we conclude this part by introducing a mixed of open issues in CA and varieties of "Challenges of Continuous Process Auditing (CPA)"

Methods for Continuous Process Auditing (Part II):

In Part II, we present in details the approaches that we proposed for accessing heterogeneous distributed data sources and for rule-based Continuous Process Auditing (CPA). Chapter 6 introduces a novel way of accessing heterogeneous distributed data sources. It also introduces an approach to represent process in a distributed environment. In Chapter 7, we presented a mechanism to construct audit rules that are defined by audit professionals and a mechanism to construct their AROPs. The development cycle (from conceptualization to construction to operation) of Hybrid Layered Ontology is also presented in the context of Healthcare system. An approach to map between Domain and Process ontologies, and Audit Rule Ontology of a Process (AROP) is also described in Chapter 7.

Evaluation and Applications (Part III):

In Part III, we introduce an algorithmic approach to the evolution our proposed ontologies and evaluate our approaches in Chapters 6 and 7. In Chapter 8, we describe an algorithmic approach to identify either conceptual or application types of changes and to edit the identified changes in three ways to linking the conceptual and semantic relations in ontologies. All three layers of ontologies - Common Ontology, Process Ontology, and AROP were encapsulated by the obtained knowledge base that described in ontology operation. Chapter 9 demonstrates the evaluation and usefulness of approaches to the Continuous Process Auditing (CPA) using three different datasets from three different pervasive environments. It also present the experiments, results and analysis of outcome. We conclude Chapter 9 with a discussion of results and concluding remarks.

Conclusion (Part IV):

In Part IV, we round up this dissertation by an overall summarization and an outlook to future work. In Chapter 10, we discuss some thoughts and directions that have arisen directly from our research. These thoughts require substantially more work beyond the scope of this dissertation; rather, we present them as problems that are open for future research opportunities. Chapter 11 summarizes this dissertation and we conclude with some final remarks.

Part I

Background and Foundations

Chapter 2

Auditing

The word 'Audit' originated from the Latin word 'audire' which means 'to hear'. According to Merriam-Webster[a] dictionary: auditing is a formal examination of an organization's or individual's accounts or financial situation.

An audit is a professional, independent opinion about a company's utilization of Generally Accepted Accounting Principles (GAAP) when preparing its financial statements. In the modern world's perspective, auditing is not just related to accounts or financial transactions. More precisely, it is a systematic, formal and disciplined process of objectively obtaining and evaluating evidence regarding assertions about economic actions and events to ascertain the degree of correspondence between those assertions and established criteria and communicating the results to interested users. In a financial audit, the assertions about which the auditor seeks objective evidence relate to the reliability and integrity of financial and, occasionally, operating information. The examination of the objective evidence underlying the financial data as

[a] http://www.merriam-webster.com/dictionary/auditing

reported is called an audit.

Auditing is governed by professional standards, completed by individuals, normally certified professionals, independent of the process being audited. Formal review of professional standards; lawful communication with recommendations and corrective action measures; systematic and structured approach involving with planning, sampling, testing and validating & documented follow-up of corrective actions are a few typical characteristics of an Audit. It can be done internally (by employees of the organization) or externally (by a recommended or certified agency).

Using straightforward terms, auditing encompasses a variety of methods used to measure the compliance of a system to defined rules and guidelines. Auditing is just one facet of a more extensive set of process, often based on accounting, but extending to support assurance that the system is working as intended. Clearly, auditing is applied in many domains, from government to business, education and health care, among many others. Underscoring the breadth of auditing applications is the fact that most auditing is still performed by human agents, professionally trained and experienced in many aspects of evidence gathering, interpretation of rules and guidelines, and clarifying of final reports in respect of limitations.

Increasingly, complex systems have grown beyond the capacities of human driven auditing to perform meaningful audits in a timely fashion that serves stakeholders and oversight bodies. Such systems encompass networks of human agents and also highly automated software systems with semi-autonomous sub-systems, all of which are assumed to be imperfect or vulnerable to risk. To this end, researchers have focused attention on the notion

of automating significant parts of auditing, both as embedded components within systems working autonomously, and as decision support components serving human analysts.

2.1 External Auditing

External auditing is the examination of an organization's financial statements by a qualified, independent accountant. The primary purpose of an external audit is to test the validity of the documents used to support the amounts and disclosures provided in an organization's financial statements. In other words, an individual outside of the organization that objectively assesses the effectiveness of the organization's quality system. Usually, periodic or specific purpose (ad hoc) audits are conducted by external, independent, qualified and/or certified accountants or auditors.

2.2 Internal Auditing

The Institute of Internal Auditors defines internal auditing[b] as, "an independent, objective assurance and consulting activity designed to add value and improve an organization's operations". An internal audit includes, but is not limited to, examining the effectiveness of an organization's risks and internal control procedures, analyzing and testing these controls, and recommending any necessary changes for improvement. An internal audit can be categorized in terms of the following three types:

[b] https://na.theiia.org/standards-guidance/mandatory-guidance/Pages/Definition-of-Internal-Auditing.aspx

2.2.1 Transactional Auditing

A transaction is any activity in business that occurs when something of value is exchanged with something else of value. Auditing procedure related to examining specified transactions and supporting documentation used by the auditor to check internal-controls reliability of an organization. As an example, a customs personnel audits to confirm the accuracy of individual import, export, and excise declarations made by traders to customs. Normally a selected number of specific transactions are tested by auditors to see if controls are performing properly and it also helps to determine the scope of audit work.

2.2.2 Process Auditing

A process audit is used to verify that processes are operating within specified limits and achieving targets or objectives. Process audits examine one or more processing steps. A process audit is an evaluation of the sequential steps and interactions of a process within a system. Process auditing provides value by evaluating processes, their controls, risks, and the achievement of objectives. They may also provide information on the ability of the process to produce a quality output. A properly done process audit is much more than verification that processing steps are being followed.

Auditors and management can benefit by conducting process audits and using process techniques to better test and evaluate system controls. Needs for process auditing and its uses, opportunities, technological support and existing & potential new methodologies are more elaborately discussed in Chapter 4.

2.2.3 Continuous Auditing

In Internal Auditing, there are multiple ways of performing audit continuously for various auditable tasks and controls. We divided the whole continuous auditing, mainly in four categories: Mathematical Continuity, Real-time, Procedural, and Functional. In the aspect of implementation, it is a combination of Continuous Data Assurance plus Continuous Control Monitoring [6].

We will continue this discussion of continuous auditing (CA) in the form of taxonomy and the necessity of continuous process auditing (CPA) in the next two Chapters 3 and 4, respectively.

2.3 Service-Oriented On-Demand Auditing

In a service oriented computer audit system, agent technologies are used and a set of components, which have logical functions and are interactive as well as autonomous, are integrated together to form a system framework. The majority of service oriented audit systems adopt the pattern of auditing on the spot, performing the collection of audit data based on data file exchange systems and local area network transport systems. Service oriented audit systems can perform specific functions and are adopted in the audit process. As an assistant tool, a computer audit system can perform such tasks on audit data as collecting, cleaning up, computing, counting, querying and generating report, providing the actual auditor, who also performs all kinds of examinations and collects audit proof, with help accessing various systems and using appropriate technologies.

Today, web-based technologies extend a company's internal systems into

the external environment that exchanges data over secure networks, then gets auditing services under Service-Oriented Architecture (SOA) environment techniques when the service is needed. This is also called on-Demand auditing.

Chapter 3

Taxonomic Analysis of Continuous Auditing

An audit is a professional, independent opinion about a company's utilization of Generally Accepted Accounting Principles (GAAP) when preparing its financial statements. In the modern perspective, auditing is a systematic, formal and disciplined process of objectively obtaining and evaluating evidence regarding assertions about economic and other kind of actions and events to ascertain the degree of correspondence between those assertions and established criteria and communicating the results to interested users. Thus, modern auditing is much more than just relating to accounting or financial transactions.

3.1 Introduction to Continuous Auditing

Continuous Auditing (CA) is a type of auditing which produces audit results simultaneously with or closely proximate to the occurrence of relevant events. According to *The IIA's Global Technology Audit Guide (GTAG) Continuous Auditing: Implications for Assurance, Monitoring, and Risk Assessment*, continuous auditing is defined as the automatic method used to perform control and risk assessments on a more frequent basis. Technology plays a key role in continuous audit activities by helping to automate the identification of exceptions or anomalies, analyze patterns within the digits of key numeric fields, review trends, and test controls, among other activities.

The most common definition used for CA, as proposed by the 1999 CICA[a]/ AICPA[b] committee is: "A continuous audit is a methodology that enables independent auditors to provide written assurance on a subject matter, for which an entity's management is responsible, using a series of auditors' reports issued virtually simultaneously with, or a short period of time after, the occurrence of events underlying the subject matter" [56, 57]. Most up-to-date version of the CA definition is discussed in Section 4.1 of Chapter 4.

By analyzing the above two definitions, we note that Continuous Auditing is a methodology or framework that enables auditors to provide written assurance on the subject matter using one or a series of reports issued simultaneously. They are usually technology driven and designed to automate error checking, detecting anomalies, finding misconduct and data verification,

[a]CICA - Canadian Institute of Chartered Accountants. As of January 01, 2013, Chartered Professional Accountants of Canada (CPA Canada) was established by CICA and CMA Canada www.cpacanada.ca

[b]AICPA - American Institute of Certified Public Accountants www.aicpa.org

and so on, in real time, then they generate alarm triggers for appropriate stakeholders. Therefore, continuous auditing is designed to enable auditors to report on subject matter within a shorter period of time than under the traditional auditing practices.

Although there are many definitions of CA, it is the constant collection of evidence as it occurs and prompt evaluation of the collected evidence which identify the primary aspects of CA [48]. There are two facets common to all definitions: i) aim to provide any sort of assurances and ii) the evaluation reports are produced as soon as possible after the evidence occurs and is collected. CA is suggested to only internal auditors but Alles et al. 2004 [6] referred to applying CA to the external audit process. ISACA Standards describe that CA may be viewed to encompass both internal and external auditing.

In order to produce evaluation reports promptly, it is desirable that a CA system should be a highly automated electronic system. A CA system must rely on feeding the reliable flow of information or evidence in a fully automated process. Thus, information technology plays a vital role in the processing of reliable information continuously [8] in continuous auditing.

One of the main purposes of CA is to assist auditors in providing assurances on collected evidence or information by verifying evidence or information integrity [29]. Technology must be implemented in order to collect accurate and reliable evidence and information. Therefore, demands for Continuous Auditing and Assurance are originated from the organizations' necessity and requirement for more reliable and relevant information. Basic economic changes in the industry and the intrinsic nature of controls over data and transactions;

right now different industry require to ensure wave of disclosures, set of legal procedures and improper accusations; increased risks of human and system instability; internal rules in the industry and legislative enforcement (Section 404 of the Sarbanes-Oxley Act 2004) to improve reporting accuracy and transparency; adapting to the new technologies to cope up with the competition; and finally increasing the efficiency and profitability are the few major facets of CA to be required and hence be implemented in any organization.

3.2 Previous Works

The Embedded Audit Module (EAM) approach introduced by Groomer et al. in 1989 that a series of auditor-developed master files are instantiated in the live client system and test transactions are entered as desired by the auditor. But the Continuous Auditing methodology first introduced at AT&T Bell Lab in 1989 and then again in 1991 by Vasarhelyi and Halper [80]. Bell developed a Continuous Process Auditing System (CPAS) for the internal audit organization to deal with the problem of auditing large paperless database systems. The system was just monitoring the databases rather auditing continuously. In 1999 Kogan et al. [48] viewed CA as a type of auditing that produces audit results simultaneously with, or a short period of time after, the occurrence of relevant events.

In 1999 *CICA and AICPA Research Report on Continuous Auditing* concluded that continuous audits are viable under certain conditions and that automated "alarm triggers" would be needed to signal anomalies and errors. They called for research to show how auditors could effectively use sophis-

ticated automated audit tools. Enabling new emerging technologies, surge in corporate scandal (i.e. Enron) & malpractices, detection of security and technological risks & human errors in a Process, and periodic adaptation to new legislative acts and regulations have put currently "Auditing" at a critical point. Specifically, passage of the Sarbanes-Oxley Act (SOX) in 2002 imposed sweeping changes on publicly traded companies and the accounting profession. SOX emphasized that assurances about internal control practices and operations.

The technological aspects of CA is the primary concern prior to continuous auditing research. Prototypes CA system design were introduced by Groomer and Murthy in 1989, Vasarhelyi and Halper in 1991, Woodroof and Searcy in 2001 and Santos et al. in 2008[33, 80, 87, 66]. Alles et al. 2002[5], Razaee et al. 2002[65] and Flowerday et al. 2006[28] have discussed the CA enabling technologies whereas Dull et al. 2006[25], Alles et al 2008[8] and Pedrosa et al. 2012[62] discussed in the context of Computer Assisted Audit Tools (CAAT) and Techniques in various form of real world applications. Directions for future research in CA have given in Kogan et al. 1999[48], Elliott 2002[27] and Brown et al. 2006[17] and research studies on other aspects of CA have emerged (Kuhn et al. 2010[49], Omoteso et al. 2010[59], Chan et al. 2011[24], Vasarhelyi et al. 2012[79, 78]) recently.

A social & economic perspective of CA and a feasibility study were addressed in Alles et al. 2002[5]. A behavioral perspective from the perceptional view-point of management was discussed Searcy et al. 2003[67]. Vasarhelyi et al 2004[77], Verschoor 2006[84], Shing-Li et al. 2007[51] provide various theoretical and practical framework and/or model on the use of analytical

monitoring in CA. Bovee et al. 2005[16] have introduced the agent based reporting and auditing framework with net knowledge and XBRL. Alles et al. 2006 describe a pilot project CA system at Siemens to audit business process controls and then used the experiences gained at Siemens on the implementation of CA in Brazil[11] in 2006.

At the end of 2012, AICPA discussed the evolution of auditing in the perspective of future audit[19] and also discussed the current state of CA and continuous monitoring with the intent of offering companies insight[20] in their two white papers.

3.3 Categorical Review of Internal Auditing

Researchers have explored the concepts of continuous auditing mainly in internal auditing. Based on well-practiced trend in the financial industry, internal auditing can be distinguished in three different sub-groups: periodic, continuous and mixed of both continuous and periodic. Figure 3.1 depicts the principles of auditing and its sub-groups.

3.3.1 Periodic Auditing

A process audit is used to verify that processes are operating within specified limits and achieving targets or objectives by examining one or more processing steps. It is an evaluation of the sequential steps and interactions of a process within a system that provides value by evaluating processes, their controls, risks, and the achievement of objectives.

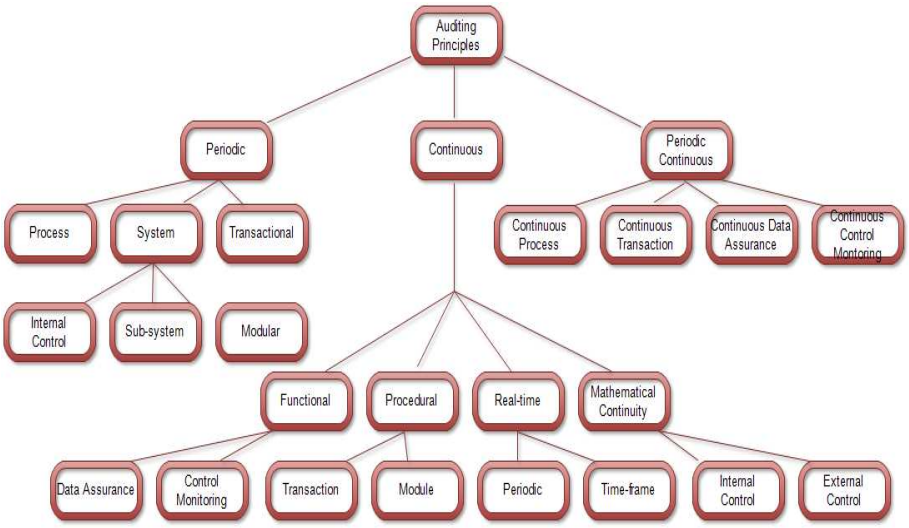


Figure 3.1: Taxonomy of Auditing Principles

Process:

A process audit is an evaluation of the sequential steps and interactions of a process within a system. Fig. 6.3 depicts the discovery of a process in a system. Process auditing provides value by evaluating processes, their controls, risks, and the achievement of objectives. They may also provide information on the ability of the process to produce a quality output. A properly done process audit is much more than verification that processing steps are being followed.

Transactional:

A transaction is any activity in business that occurs when something of value is exchanged with something else of value. Auditing procedure related to examining specified transactions and supporting documentation used by the auditor to check internal-controls reliability of an organization. As an exam-

ple, a customs personnel audits to confirm the accuracy of individual import, export, and excise declarations made by traders to customs. Normally a selected number of specific transactions are tested by auditors to see if controls are performing properly and it also helps to determine the scope of audit work.

System or sub-system (module):

It is an approach to auditing. A system audit to assess the internal control system of an organization, possibly, to assess the quality of many different kind of system or sub-system (i.e. the quality of accounting system and the level of testing required from the financial statements; the process of collecting and assessing evidence of Information System to show that safeguards to protect against abuse, safeguards assets maintains data integrity and allows the organization to continue successfully) to determine the organization's system is adhering or able to adhere to regulatory requirements. A System could be constituted with one or more modules (sub-systems).

Periodic auditing can be done internally by the organization's auditing and monitoring team or externally by the industry accepted certified auditing professionals.

3.3.2 Continuous Auditing

Continuous auditing is defined as the automatic method used to perform control and risk assessments on a more frequent basis. Most common definition as proposed by the 1999 CICA (currently CPA Canada)/AICPA committee: "A

continuous audit is a methodology that enables independent auditors to provide written assurance on a subject matter, for which an entity's management is responsible, using a series of auditors' reports issued virtually simultaneously with, or a short period of time after, the occurrence of events underlying the subject matter." By analyzing above two definitions, we see that Continuous Auditing is a methodology or framework that enables auditors to provide written assurance on the subject matter using one or a series of reports issued simultaneously. Technology plays a key role in continuous audit activities by helping to automate the identification of exceptions or anomalies, analyze patterns within the digits of key numeric fields, review trends, and test controls, among other activities.

Mathematical Continuity:

An auditing objective that will be automatically done based on the continuity of one or more auditing related functionality/elements/controls.

Real-time:

After completion of (and/or within) the certain period of time an auditing objective to be completed/detected automatically. How short time-frame could be is the mostly considerable components for this type of auditing. Depending on the industry, system and environment time-frame could be from min (i.e. Blood pressure machine) to sec (i.e. heart monitoring device at an intensive care unit) and even a nano-sec (i.e. process function of an atomic plant).

Procedural:

Procedural auditing normally consists of several sequence of steps that include preparation, conduct and completion of a task, which are usually administered by a continuous audit manager. Each step can be a sequence of activities and each activity is a sequence of actions. The sequence of steps is critical to whether a statement or document is a procedure or something else. Auditors always know about these sequence of steps to better monitor the continuous procedure. They also provide results either as a form of recommendations or assurance for its improvement, if needed. Continuous procedural audit is also part of continuous monitoring that ensures the organization's procedures operation effectively.

Functional:

To assure the business is running effectively and efficiently, management team normally monitors some business function continuously. Functional audit does not provide independent assurance of a function for the stakeholders rather it does assure the smooth and effective running of a business function which is also part of continuous monitoring.

A major challenge of implementing *continuous auditing* systems is combining Continuous Data Assurance (CDA) and Continuous Control Monitoring (CCM).

3.3.3 Periodic Continuous

Continuous Process auditing:

A process audit is, elaborately, an evaluation of outcomes to determine whether the activities, resources and behaviors are being done and/or managed in an effective and efficient way. It is not simply following a path through the departments from actions to outcomes - which is actually a form of transition audit. Since a process always generates outcomes/results, therefore it may be subject to a process audit. Now the question is - whether process auditing should be done in a traditional way or continuously? Continuous process audits evaluate the results/outcomes of a process continuously whether or not they are being generated by an effectively completed/managed process. It can be achieved either by auto-generated triggers or by a designated software tool to perform the audit in a specific time-delay or upon the completion of a process. One of the early models implemented that formed the basis for the later models was the *Continuous Process Auditing System* (CPAS), which was developed at AT&T Bell Laboratories for internal auditing of large 'paperless' real-time systems [80].

Continuous Transaction auditing:

In financial auditing, vouching is a term that refers to the inspection of documentary evidence supporting and substantiating a transaction. The objective of establishing the authenticity of the transactions recorded in the primary books of account is followed in a transactional audit. Vouching is considered as the backbone of auditing in the financial industry. As transaction-based

industries/institutions emerging to the evolving technologies, transaction auditing performs automatically on a continuous basis.

Continuous Data Assurance:

Continuous Data Assurance (CDAssurance) verifies the integrity of data flowing through the information systems. Continuous Assurance in audit practices is moving towards the maximum possible degree of audit automation as a way of taking advantage of the technological basis of the modern systems for analysis at the transactional level to provide more detailed assurance and increase audit automation. The development of Continuous Assurance requires a fundamental rethinking of all aspects of auditing, from the way in which data is made available to the auditor, to the kinds of tests the auditor conducts, how alarms are dealt with, what kinds of reports are issued, how often and to whom they are issued, and many other factors, the importance of some of which will only become apparent as Continuous Assurance is implemented. It is important for the profession and other stakeholders to start thinking about the impact of Continuous Assurance on auditing now, when it is easier to put in place the foundations for this change, rather than when technologies and practices have already become established [6, 79] in a piece-meal fashion. Continuous Data Auditing (CDAuditing) is also a subset of tasks under the hood of Continuous Data Assurance [47]. Though auditing professionals interchangeably used the terms CDAssurance and CDAuditing for same the purpose but CDAuditing, that verifies the data integrity, is part of CDAssurance that provides not only the data integrity assurance to appropriate stakeholders and auditors but also assures data availability and robust

of data flowing mechanisms through the EIS.

Continuous Control Monitoring:

The Institute of Internal Auditors described Control Monitoring as the Independent control review mechanism to help organizations assure the effectiveness of Internal controls, reduce operational risks, minimize profit erosion, and mitigate the risk of fraud, while meeting regulatory requirements. Adhoc control progresses from basic level in the direction of audit contribution, and analytic sophistication in a linear way to the level of applied, managed, automated, and in the final stage it becomes Continuous Control Monitoring. There are a few vendors already providing to industry specific kinds of Continuous Control Monitoring services [22] or software tools[c]. Sometimes their interpretation of these kind of tools or services are attributed to Continuous Auditing; but, they all lack basic fundamental aspects of Continuous Auditing [65, 4].

Continuous Risk Monitoring and Assessment:

Continuous Risk Monitoring and Assessment (CRMA) is used to dynamically measure risk and provide input for audit planning [24]. CRMA is a real-time integrated risk assessment approach, aggregating data across different functional tasks in organizations to assess risk exposures [1] and provide reasonable assurance on the firms risk assessments [78, 20, 19].

[c] http://www.protiviti.ca/en-CA/Pages/default.aspx

Chapter 4

Continuous Process Auditing (CPA)

Most processes are unique to the organization which employs them and they are defined by either the management team or the internal auditing team of the organization. This point differentiates between process definitions localized to an organization as opposed to those definitions that apply to "off the shelf" process components (e.g. technology components). A process that is being audited continuously with the help of technology underscores the applicative sense of Continuous Process Auditing (CPA). The concept of CPA was introduced by Vasarhelyi and Halper in 1991 at AT&T Bell Laboratories for internal auditing of large 'paperless' real-time systems [80]. Continuous analytical monitoring of business processes [10], process based auditing [12] and conceptual model of web-based online auditing [2] have been introduced and in practice for several years now.

Process based auditing of all the four types of continuous auditing that were discussed in Section 3.3.2 are commonly known as Continuous Process Auditing (CPA). It is the Compliance and operational part of Continuous Process Auditing that is our prime interest and addressable problems are identified and defined throughout Chapter 4. However, we begin by distinguishing what are the similarities and differences between Continuous Auditing (CA) and Continuous Monitoring?

4.1 Continuous Auditing vs Continuous Monitoring

By analogy with conventional auditing, continuous audit procedures can be designed either to test internal controls (continuous control monitoring) or to execute substantive testing (continuous data assurance). From the procedural point of view, Alles et al. 2008 [9] divides continuous auditing into two distinct aspects:

> Continuous Auditing (CA) = Continuous Control Monitoring (CCM)
> + Continuous Data Assurance (CDA)

The essence of both Continuous Auditing and Continuous Monitoring is to provide a radically improved level of assurance on operations. There are similarities between them, but they are not quite the same processes. Continuous audit performs auditing activities on a frequently repeated schedule to provide ongoing assurance and more timely insight into risk and control issues. On the other hand, continuous monitoring is essentially a process that fall under

the management's responsibility, in which key business process transactions and controls are constantly assessed. This permits ongoing insight into the effectiveness of controls and the integrity of transactions running within them.

Though both processes tend to produce similar outcomes, the primary difference between them is related to ownership of the process. Continuous audit is owned by the audit function and can include any audit process that is repeated regularly, whereas management owns the continuous monitoring process. Monitoring the effectiveness of controls systems is the management's primary responsibility that benefits them by getting timely insight into transactions that are the result of fraud, error or abuse.

Deloitte published a whitepaper [63] in 2010 on how *Continuous Auditing* idea to implementation. They have provided a differential list between continuous auditing and continuous monitoring as shown in Table 4.1.

Vasarhelyi, Alles, and Williams [79] suggested the addition of Continuous Risk Monitoring and Assessment (CRMA) into the CA schema where: $CA = CDA + CCM + CRMA$. The assurance process must be coherent with the components of CA to be useful and effective. Figure 1-4 of [35] expands Vasarhelyi, Alles, and Williams components to add an element of compliance monitoring, expanding the scope of the CA and CM effort. The authors also contend that CRMA should ideally be fully integrated within a structure that includes both Continuous Controls Monitoring (CCM) and Continuous Data Assurance (CDA) such that a robust system of continuous auditing is ultimately achieved [82].

The Compliance Monitoring (COMO) approach (see page 11-12 of [35]) that would create comprehensive taxonomies of compliance issues and pro-

Continuous Auditing (CA)	**Continuous Monitoring (CM)**
Ongoing process that enables *internal audit* to:	Ongoing process that enables *management* to:
Collect from processes, transaction, and accounts data that supports internal and external auditing activities	Assess the effectiveness of controls and detect associated risk issues
Achieve more timely, less costly compliance with policies, procedures, and regulations	Improve business processes and activities while adhering to ethical and compliance standards
Shift from cyclical or episodic reviews with limited focus to continuous, broader, more proactive reviews	Execute more timely quantitative and qualitative risk-related decisions
Evolve from a traditional, static annual audit to a more dynamic plan based on CA results	Evolve from a traditional management plan to a more dynamic plan based on monitoring outcomes
Reduce audit costs while increasing effectiveness through IT solutions	Increase the cost-effectiveness of controls and monitoring through IT solutions

Table 4.1: Difference between Continuous Auditing and Continuous Monitoring

gressive updates for regulatory changes acknowledged by geography, area of activity, and the nature of compliance rule (qualitative, quantitative, mixed, or other). This would reformulate the definition of the CA to: $CA = CDA + CCM + CRMA + COMO$ which would open the doors for the potential usage of an existing and conceptual IT platforms.

4.2 Necessity of CPA in Compliance and Operational Audit

For Compliance Audit: Compliance Control and Monitoring enhance capabilities for ensuring various standards and compliances across physical and virtual components like data centers, databases, security monitoring devices etc.

Policy: automating policy definition and policy life cycle management with new contents and new policy, automating the assets to controls mapping, periodic delivering and updating the content and standards.

Risk: aligning compliance operations and security with defined risks, prioritizing resource allocation according to risk factor.

Managing expensive and labor-intensive compliance and audit requirements. Determining the appropriate set of IT policies and controls to manage IT risk and compliance. Consistently implementing and enforcing IT policies and controls. Quickly responding to IT auditors and other management requests to demonstrate compliance. Determining vulnerabilities based on the security policy, compliance, and risk management.

For Operational Audit: To evaluate the operational activities of a given company or other organization that gives much more understanding and in-depth review of the business processes. To verify the components of the audit and the associated concerns with in control of high risk areas and control activities.

Data auditing and analytics: In terms of data dimensions such as volume, velocity, variety and veracity - there are in need of operational data process

auditing in consistency, integrity, identification, aggregation and confidentiality.

Control matrix: validation of control matrix to improve the better in control of whole operational process. Continuous validation of test procedures for each key control and automated report of the findings.

Opinions: Generating real-time opinions of components' behavior to make better in-time decision to improve over all control systems as well as the operational processes.

Rule based operational audit: Reaction RuleML[a] for action and/or reactions. Reaction rules subsume Complex Event Processing (CEP) and Knowledge Representation (KR) rules, as well as Event-Condition-Action-Postcondition (ECAP) rules. ECAP rules specialize to Event-Condition-Action (ECA) rules, which themselves specialize to Condition-less Trigger (EA) rules and to the rule subfamily of Event-less Production (CA) rules.

[a] http://reaction.ruleml.org/

Chapter 5

Existing Technologies and Approaches

In the last decade, researchers around the world have explored the characteristics of continuous auditing system and have provided the intermediate guidance for implementation of continuous systems. In this section, we have categorized the characteristics of auditing systems.

The need to implement and establish the efficacy of evidence-based continuous auditing modalities is of concern to all who support and encourage this field of endeavor. Research has been conducted in the areas of continuous auditing with various aspects and different approaches. Since Trust services frameworks are mostly part of the AICPA/CICA's assurance services to evaluate the auditing standards changes, we have not included any third party framework except those that are available from the AICPA/CICA.

5.1 Continuous Auditing System Models and Frameworks

5.1.1 Theoretical/Conceptual Research Framework:

Woodroof and Searcy [87] has presented a prototype of their theoretical model that is limited to a specific domain to continuously monitor whether actual values of a client's variables are in compliance with standards for these variables set out in the debt covenant agreement. They also presented the theoretical mechanism and characteristics of a reliable and a secure continuous auditing system that compliance with AICPA/CICA standard in a continuous audit environment.

Elliot [27] stated the possibilities and future aspects of continuous assurance and auditing. Since continuous reporting is major part of the continuous assurance that encouraged him to present an insurance (assurance) transaction model that insured the payment for losses caused by faulty information flow.

Alles et al. 2002 [5] theoretically presented the idea of the components of continuous assurance. They identified the three essential components of assurance are i) capturing transactional data, ii) monitoring and analyzing it, and iii) communicating the outcome of the analysis. They emphasized the importance of understanding that continuous assurance implies for each of these steps.

A "Black Box (BB) log file" concept was presented by Alles et al. 2004 [7] that is a read-only, a third-party-controlled record of the actions of audi-

tors, especially in regard to their interactions with management and choice of audit metric and models. This concept aims to be more accurate and more supportive of the management process in real-time detection of fraud in continuous assurance systems. Their idea was to integrate the BB logging in the last steps of generic continuous assurance methodology that may produce statuary reports to the appropriate agencies that control the several layers of monitoring.

In 2008, Carlos Santos et al. [66] presented a conceptual model for Continuous Organizational Auditing with Real Time Analysis and Modern Control theory. The process of evaluating and validating is one of auditing which, in order to be performed according to good practice and should be based on an internal control system. Two components scientifically conceptualize in this model were 1) the consistent and coherent design of an internal control system based on modern control theory, and 2) the formal verification of the rules that make up the internal control system's specific aims; a conceptual model able to support continuous organizational auditing using real time analysis. Table 5.1 summarizes characteristics of all models and frameworks.

5.1.2 Generalized Approach:

Razaee [65] described a generalized framework as a continuous auditing approach that enables auditors to provide some degree of assurance on continuous information simultaneously with, or shortly after, the disclosure of the information. They stated that standardization of data (diverse file types and various record formats produced by various systems and sources) is one of the

Characteristics	**Theoretical/Conceptual Models and Frameworks**
Continuous Monitoring	**Woodroof and Searcy**'s model is limited to compliance standard to monitor actual value from client. **Alles et al. 2002 and 2004** presented the monitoring is part of assurance and analysis.
Continuous Assurance	**Elliot** stated the insurance (assurance) transaction model that insured the payment loss from data loss. **Alles et al. 2002** presented component based methods for continuous reporting assurance whereas they introduced, in 2004, a generic continuous assurance concept called "black bog" logging for statuary report production.
Real-time Detection	**Carlos Santos** et al. [66] introduced real time analysis in the process of validating and evaluating in auditing internal control and **Alles** et al. [7] aims to detect fraud in real time in continuous assurance of management process.
Theory and Concepts	**Carlos Santos** designed the internal control system using Modern Control theory and the formal verification of the rules of internal control's specific aims. **Alles** in 2002 used the three component concepts of capture, monitor & analyze data, and communicate outcome and in 2004, introduced the concepts of choice of 'audit metric' and 'black box' logging of record of actions.

Table 5.1: Characteristics of Theoretical/Conceptual Models and Frameworks

most complex and challenging aspects of building continuous auditing capability. This model is depicted a generic solution to continuous auditing that seems to emphasize on the standardization of data.

Onion's proposed model for secure continuous auditing [60] introduces the concept to guarantee the integrity of captured accounting information to monitor all keystrokes and transaction within the system, and then search for patterns in groups of transactions by using Expert systems. The keystroke level data examination involves monitoring database utilities and applications

for commands which could cause fraud or error. They defined a XML-based generic schema, using eXtensible Continuous Auditing Language (XCAL), for a transaction would allow one expert system to find through the data mining of the keystroke level data examination. This conceptual model is depicted a generic solution, to secure data integrity, that attempts to find a solution by using expert systems where a various data formats is available.

Woodroof and Searcy [87] proposed a domain specific model for debt-covenant compliance. They seems to be working on a prototype of continuous secure-reliable auditing system that assures agreement between all involved parties and dynamically produces audit reports to the user through Web within the continuous auditing environment. This model also enabled the Web services and digital agent technologies for the first time in continuous auditing system. (See Table 5.2 for summery)

5.1.3 Service Oriented Approach:

Murthy and Groomer in 2004 [55] presents a web service based model that relying on a number of components of Web services technology for continuously audit business processes. It is referred to as continuous auditing web service (CAWS) and this CAWS mechanism would run as a "web service" in the audit firm's computing environment and could be applied at a very granular level to provide assurance about specific business processes, at a very aggregate level for providing assurance relating to continuously reported earnings, or to provide continuous assurance (CA) about the operation of internal controls resident in the audit client's environment. This framework and

Characteristics	Generalized Models and Frameworks
Data Standardization	Razaee [65] tried to attack the problem of heterogeneous data and file formats by standardizing data. Onions [60] also finds the solution to secure data integrity by defining XML-based generic schema for various data formats.
Specialized System	Woodroof and Searcy's [87] proposed frameworks as a web service using web-based and agent technologies to provide data assurance between two parties. Onions approach of expert system, through the data mining of keystroke level data, to tackle the various data formats issue.
Domain Specific	Woodroof and Searcy's main focus of frameworks was on debt-covenant compliance with standards set out in the debt agreement. Razaee and Onions both emphasized on the data standardization domain to find a generic solution for continuous auditing.

Table 5.2: Characteristics of Generalized Models and Frameworks

technologies facilitate a Web-service-based continuous auditing mechanism in an XML-enhanced world.

The Service-Oriented Architecture (SOA) based model for continuous auditing for online information system proposed and carefully studied by HuanZhuo Ye et al. [89] in 2008. At present, the research of continuous auditing faces a series of problems, such as the accuracy of data collection, real time, comprehension and flexibility of audit. Applying SOA to the auditing system will help to solve these problems.

5.1.4 Trust Services Framework:

In 2000, the AICPA and the CICA have jointly created the "SysTrust", a new assurance service, to evaluate the need for auditing standard changes for

Models and Frameworks	**Characteristics of Service Oriented Models or Frameworks**
Service Oriented	**Murthy and Groomer in 2004** [55] modeled a continuous auditing mechanism as a *web service* using a number of web technologies that provides *continuous assurance* about the operation of *internal controls*. **HuanZhuo Ye et al.** [89] in 2008 conceptualized a *Service-Oriented Architecture (SOA)* based model for continuous auditing to solve the *accuracy of data collection* for online information system in *real time*.

Table 5.3: Characteristics of Service Oriented Models and Frameworks

the advent of more CA methodology. A system earn an unqualified SysTrust report by meeting the followings:

- Principles: the four essential principles underlying reliable systems are availability, security, integrity and maintainability.
- Criteria: there is a set of criteria, for each principle that enables to assess whether a system has achieved that particular principle.
- Engagement: in the US, it is performed under AICPA *Statement on Standards for Attestation Engagements* whereas it is the *CICA Handbook* in Canada.

The AICPA and CICA also developed a new assurance service, in the same year as SysTrust, called WebTrust to promote confidence and trust between consumers and companies conducting business in the Internet that relies on a series of principles and criteria as SysTrust. As of Sept 15, 2009 - Trust Service Principles and Criteria has been revised to adopt 'confidentiality and privacy'

as two more principles for both SysTrust and WebTrust. A former fourth principle Maintainability and its related criteria and illustrative controls have now been folded into the principles for 'availability, security, and processing integrity'. The trust services principles and criteria of security, availability, processing integrity, and confidentiality are organized in these four areas: *polices* relevant to the principle, *communicated* its defined policies to responsible parties and authorized users of the system, operation *procedures* to achieve its objectives in accordance with its defined policies, and *monitoring* the system and taking action to maintain compliance with its defined policies.

Similar kind of service also provided by Statement on Auditing Standards (SAS) No. 70. It is a service organization and is a widely recognized auditing standard developed by the AICPA. A "SAS 70 audit engagement" provides a report on a service organization's controls related to financial statement assertions of the user organizations. Its only for systems that provides transaction or data for the user organization whereas SysTurst for any kind of system.

5.2 Comparative Analysis of Continuous Auditing Methodologies

The Assurance Services Executive Committee of the AICPA evaluates the need for auditing changes for the advent of a more CA methodology. This committee jointly with CICA issues the Trust Services Principles and Criteria for CA methodology. Alles et al. 2004 [7] stated the following 7 general characteristics of Continuous Auditing methodologies based on those Trust

Services Principles and Criteria:

1. a layer of software (aimed at process control and monitoring) on top of most critical corporate software systems

2. an instantiation of the control and monitoring process aimed at business process assurance by both internal and external assurors

3. a constant stream of measurements (metrics) engineered out of key processes

4. a sophisticated dynamic set of standards (models) to compare with the metrics

5. a set of dynamic exception metrics to determine when an alarm is to be issued, and its degree of importance

6. an analytic layer to perform additional analysis related to several corporate functions (auditing, fraud evaluation, accounting rule compliance, estimate review) and

7. a new level of statutory reporting that may include reports to governmental agencies.

There are in need of research to device methodologies to be implemented in tools and techniques to fulfill these above characteristics. Following subsection discusses the existing tools and techniques in development and research of the methodologies towards the Continuous Auditing.

5.2.1 Research Methodologies in Practice:

- *Continuity Equation (CE)*: it is commonly used in physics as mathematical expressions of various conservation laws, such as the law of the conservation of mass: "For a control volume that has a single inlet and a single outlet, the principle of conservation of mass states that, for steady-state flow, the mass flow rate into the volume must equal the mass flow rate out[a]".

  Mathematically [86], let quantity in transport be represented by a scalar variable, q, and let the volume V density of the quantity be ρ, and the union of all surfaces be denoted by S. Then ρ is ratio of two infinitesimal quantities:

  $$\rho = \frac{dq}{dV}$$

  where dimension is [quantity][L]$^{-3}$ and L is length.

  Continuity Equation model was first demonstrated by Wu et al. [88] and Alles et al. [10] in 2006. They constructed the CE model in form of the Simultaneous Equation Model (SEM) and the Multivariate Time Series Model (MTSM) from a Business Process auditing approach. Both the constructed models were applied in Continuous Auditing to detect anomalies in Business Processes. Later in 2010, Kogan et al [47] integrated the CE models (e.g. SEM, LRM, VAR and GRACH) with Analytical Procedures for Continuous Data Level Auditing.

- *Benford's Law*: states that the first digit in many types of data sets are distributed in a non-uniform way. In fact, this law, also called the

[a] http://nuclearpowertraining.tpub.com/h1012v3/css/h1012v3_33.htm

first-digit law, says that the number 1 will appear as the first digit about 30% of the time and the number 2 will appear as the first digit about 18% of the time, whereas the number 9 will only appear first about 5% of the time. This law has been found to apply to a wide variety of numeric data sets, including stock prices, electricity bills, population numbers, street addresses, lengths of rivers, death rates, physical and mathematical constants, and processes described by power laws (which are very common in nature). It tends to be most accurate when values are distributed across multiple orders of magnitude.

Mathematically, a set of numbers is said to satisfy Benford's Law if the leading digit d (in decimal system $d \in 1, ..., 9$) occurs with probability (P). The probability distribution of first digits can be extended in any base, in fact, $b \geq 2$. So the general form of Benford's law is:

$$P(d) = \log_b(d+1) - \log_b(d) = \log_b\left(1 + \frac{1}{d}\right)$$

Benford's law has been applied in the financial industry for long back to predict the prices and numbers. Durtschi et al. [26] in 2004, first discussed the effectiveness of using Benford's Law to assist in detecting fraud in accounting data which conform to the Benford distribution. In 2011, Bhattacharya et al. [15] uses Benford's Law as a useful classifier in a genetically optimized artificial neural network in segregating naturally occurring numbers from those that are made up. Also in 2011, Silva and Carreira [68] uses the Benford's law to highlight the most suspicious records in a data set for audit targets by identifying the subset of nonconforming records.

Research and investigation are being conducted by researchers in various institution on using Benford's Law especially in transactional data auditing and forensic accounting. Still there are huge opportunity on how to use this wonderful methodology in the field of auditing to detect suspicious human behavior.

- *Zipf's Law*: the probability of occurrence of words or other items starts high and tapers off. Thus, a few occur very often while many others occur rarely. The basic concept of Zipf's Law is that the frequency of the word occurrence in an article in fact furnishes a useful measurement and hence, management of word significance: The product of frequency of the use of words, f, and the rank order, r, is approximately constant. Many scholars believe that Benford's Law is a special case of Zipf's Law.

  In the English language, as an example the frequency of words, N is the number of words in the English language, k be their rank. s be the value of exponent characterizing the distribution, is 1 for classic version of Zipf's Law. Zipf's Law predicts that out of population of N words, the frequency of words of rank k, $f(k; s, N)$ will then be the fraction of the time the kth most common word occurs:

$$f(k; s, N) = \frac{1}{k^s H_{N,s}}$$

  where $H_{N,s}$ is the Nth generalized harmonic number. The sum of all relative frequencies in a Zipf distribution is equal to the harmonic series, $\sum_{n=1}^{\infty} \frac{1}{n} = \infty$. Word frequencies, in human languages, have a very heavy-tailed distribution, and can therefore be modeled reasonably well by a Zipf distribution with an s close to 1 [86].

Huang et al. [39] in 2008 discussed a fraud detection mechanism on the basis of Zipf's Law. They devised a technique such as a generating distinctive pattern using Count and rank steps for Zipf's analysis to identify any potential fraud records in enormous amount of data sets.

There are still tremendous potential to use Zipf's Law especially in textual data mining and use in the field of auditing to analyze human behavior.

- *Heaps' Law*: (also called Herdan's law)[b] is an empirical law which describes the number of distinct words in a document (or set of documents) as a function of the document length (so called type-token relation). Let V_R be the number of distinct words in an instance text size n, and it can be formulated as

$$V_R(n) = Kn^{\beta}$$

where K and β are free parameters determined empirically. With English text corpora, typically K is between 10 and 100, and β is between 0.4 and 0.6. It means that as more instance text is gathered, there will be diminishing returns in terms of discovery of the full vocabulary from which the distinct terms are drawn. We have not seen any auditing applications tried using this methodology yet. But there is potential to apply this approach to analyze for large amount of web content and also text-based database retrieval in real-time in the field of auditing for human behavior analysis, back-tracking of forensic documents and so on.

[b] https://en.wikipedia.org/wiki/Heaps_law

- *Pareto distribution*: The Italian economist Vilfredo Pareto first noticed that 80% of the land in Italy was owned by 20% of the population. Later he formulated this relation as follows: 'In any series of elements to be controlled, a selected small fraction in terms of number of elements almost always accounts for a large fraction in terms of effect'. Lorenz observed in countries of various sizes the similar shares of wealth distribution across the population groups.

  Like Pareto distribution, similar empirical dependencies given in logarithmic or power rank-size distributions are known in the natural and social sciences particularly, the Lorenz law, Benford's and Heap's laws, Lotka's and Bode's laws, Zipf's law and its generalization in ZipfMandelbrot law.

  In 2009, Stan Lipovetsky [52] discussed that a random partitioning model with estimation of two complimentary to 100% segments is applied to find the mean value and SD, or the point and interval means of the variables' product. Then two segments are found from the values of the mean product. These segments yield the quotient of the Pareto 80/20 rule, as well as two other standard 60/40 and 90/10 proportions. The model helps to understand the process of evaluation of the factors that influence managerial decision making and also generating audit opinions.

5.2.2 Potential Research Methodologies:

- *Machine Learning tools*: Many machine learning tools and methodologies still yet to apply in the auditing field. Artificial Intelligence based classifier, Neural Networking, Genetic Algorithm and various forms of Clustering are powerful machine learning tools has the immense potential to apply not only in the data auditing also in the field of transactional and process auditing in continuous fashion or might be possibility in real-time.

- *Back-Tracking by Modeling Human Behavior*: Certified auditors from accredited institutions follow the GAAS and GAAP to perform auditing procedures. At the of end auditing, they have to include a report with their opinions as to whether the audited components present fairly in all material respects the position of the company under audit. But in IT auditing or technology based auditing, back-tracking of any digital contents to find out any kind of suspicious behavior or anomalies absolutely necessary. Sometimes, it is also required to track back human behavior to make an honest neutral opinion. Modeling of complex human behaviors pertaining to process auditing using humans' social, group agents and also using agents in the form of different expert systems that generate opinions on certain human behaviors.

5.3 Open Issues

In this section, we highlight some of the most important issues and challenges in deploying and utilizing the continuous auditing methodologies and approaches as the future research directions.

Machine Learning Tools: Many machine learning tools and methodologies still yet to apply in the auditing field. Artificial Intelligence based classifier, Neural Networking, Genetic Algorithm and various forms of Clustering are powerful machine learning tools has the immense potential to apply not only in the data auditing also in the field of transactional and process auditing in continuous fashion or might be possibility in real-time.

Back-Tracking by Modeling Human Behavior: Certified auditors from accredited institutions follow the GAAS and GAAP to perform auditing procedures. At the of end auditing, they have to include a report with their opinions as to whether the audited components present fairly in all material respects the position of the company under audit. But in IT auditing or technology based auditing, back-tracking of any digital contents to find out any kind of suspicious behavior or anomalies absolutely necessary. Sometimes, it is also required to track back human behavior to make an honest neutral opinion. Modeling of complex human behaviors pertaining to process auditing using humans' social, group agents and also using agents in the form of different expert systems that generate opinions on certain human behaviors.

Accuracy of data collection: Failure to collect appropriate data produces erroneous results that jeopardizes auditing analysis and assessment. Collection of appropriate data for use in an auditing analysis is a key component

of any analysis. Addressing the issue of collecting appropriate data requires the answers of the following questions: what is the source of the data that needed for doing analysis? How should the samples be chosen? How should the data be collected? When should the data be collected? How should the data objectivity and consistency be ensured? Finally, procedures of quality control must be developed appropriately and implemented for all steps of data collection.

Access control and authorization: Many complex and heterogeneous systems and their data sources are semantically heterogeneous. An audit rule based authorization framework [32] [48] for complex distributed system and a policy based [49] access control and authorization system for autonomous pervasive environments has been developed. Access control and authorization of data in real-time and enforcing data security as highly encrypted-sensitive should be collected from various geographic locations is an essential issue need to be handled before deploying any auditing analysis.

Business Continuity: Business continuity management (BCM) in an organization has evolved into a process that identifies an organizations exposure to internal and external threats and synthesizes hard and soft assets to provide effective prevention and recovery. Thorough understanding of the wide range of threats, a continuous auditing mechanism should identify the risk of threats and which can be recovered quickly from crises to sustain little damage to their competitive position. Comprehensive and flexible continuous auditing approach can handle BCM issues still need to be investigated.

5.4 Challenges of Continuous Process Auditing (CPA)

Continuous Process Auditing still is in its infancy age while overcoming and building the capabilities over several challenges like accessibility of heterogeneous data and system components, compliance and operational auditing and control monitoring, acquisition of audit rules, evidence evaluation, predicting the remedies and providing, and knowledge engineering. A brief discussion of the above challenges are listed below:

Heterogeneous Accessibility

From the starting point to the end-point, processes travel through all or several components of systems and/or sub-systems and data sources. These systems and data sources may be heterogeneous in nature. CPA requires reliable and secure accessing to the systems and assurance of data availability while maintaining the data consistency, integrity, aggregation, identification, and confidentiality.

Compliance Auditing

To enhance the capabilities for ensuring various standards and compliances across physical and virtual components like data, databases, security monitoring devices etc., continuous compliance control auditing and monitoring is highly demanding. Automating policy definition and policy life cycle management with new contents and new policy, automating the assets to controls mapping, periodic delivering and updating the content and standards. Align-

ing compliance operations and security with defined risks, prioritizing resource allocation according to risk factor. Determining vulnerabilities based on the security policy, compliance, and risk management.

Operational Auditing

To evaluate the operational activities of a given company or other organization that gives much more understanding and in-depth review of the business processes. To verify the components of the audit and the associated concerns with in control of high risk areas and control activities. In terms of data dimensions such as volume, velocity, variety and veracity - continuous process operations of maintaining data consistency and integrity, providing data identification and aggregation, and protecting data confidentiality are staggering need for big data and data auditing. Validation of control matrix to improve the better in control of whole operational process. Continuous validation of test procedures for each key control and automated report of the findings. Generating real-time opinions of components' behavior to make better in-time decision to improve over all control systems as well as the operational processes.

Audit Rule Acquisition

Auditing using rule is a technique that frequently used by audit professional. Hand-picking audit rule is a difficult task that requires the analysis of evidence and process components. In CPA, continuous acquisition of audit rule is a challenge not only in the sense of analyzing evidence but also monitoring the whole process behavior. Defining an audit rule is another challenge need to

be taken care before acquiring audit rule. To update an existing audit rule, identification of an audit rule is a absolute necessary.

Continuous Control Monitoring

Continuous Monitoring of controls is to develop an effective information, communication, and monitoring system that will identify when the controls that are built into the system are not working within their prescribed tolerances, and then signal evaluation activities and monitor correction. Compliance (i.e. regulatory) and performance (i.e. fraud, operational inefficiencies) issues are identified and quantified pro-actively to improve controls testing and exception reporting. Its a continuous systematic approach of observing and checking that have a basis for maintaining data validity, reliability, and integrity to provide reasonable assurance. Challenges include identification, modification, development, deployment of controls for proactive mitigation of monitoring, reporting as well as automation of manual tasks.

Evidence Evaluation and Predicting Recommendations

Determining whether the audit evidence obtained is sufficient and appropriate to support the opinion to be expressed. Making sense of the evidence gathered is appropriate (relevant, reliable and valid) and sufficient to support assertions and to provide assurance. What evidence need to collect and/or retrieve, from where (what are the sources), and when (determination of the relevance and reliability)? Developing an evidence evaluation technique that is sufficient and appropriate. That means evaluation technique follows through to its logical conclusion throughout the whole process without being biased.

Predicting ethics and compliance recommendations is a challenge every EIS faces nowadays. Developing prediction tool that is appropriate and sufficient to optimize the evidence collection and evaluation procedure in the total auditing process of Continuous Process Auditing.

Knowledge Engineering

Knowledge engineering through Continuous auditing is an essential features that will help to transform implicit knowledge into explicit knowledge. Ontology is a formal, explicit specification of a shared conceptualization. Domain specific concepts and relations are formally defined by axioms and definitions, and there is a mechanism to organize the concepts by means of relationships, which might be hierarchical or non-hierarchical. Continuous Auditing can be utilized to engineer the domain specific implicit knowledge then transform to explicit knowledge. Detecting evidence, reporting and recommendations are the components of any auditing. These three components generate and collect several different kind of feedbacks and experiences. Modeling of feedback and experience analysis process is absolute necessary that seeks the capitalization of experimental element available in the organization. Feedback and experience based continuous auditing system can increase the efficiency and/or effectiveness to solve the particular types of problem scenarios as well as can an aid to compliance and operational system.

Part II

Methods for Continuous Process Auditing

Chapter 6

Domain Knowledge and Process Ontology

One vital element throughout auditing concerns knowledge, namely, its acquisition, interpretation and uncertainty, or vagueness, in reasoning. From the outset, auditing practice dictates that meaningful definitions must be determined and documented, for the system to be audited, components and processes within the system, evaluation measures, rules and actions to be applied, and limitations or constraints. In all aspects, auditors must work with suitable knowledge expressed in natural language terms for human consumption, but also expressed in terms appropriate for application and reasoning through logic, using computational techniques in particular.

6.1 Introduction

Extracting knowledge from text in a semi-automatic way and identifying effective procedures for achieving useful and reliable results are challenging and daunting scientific research areas [36]. Inference based on interpretations of human defined audit rules are as essential to the Semantic Web as application domain ontologies. Ontology-based reasoning has known shortcomings and limitations compared with rule-based reasoning [38]. To represent inferential knowledge, ontology alone is insufficient [37], but inferential rules are an essential part of the knowledge in an audit rule ontology for a process or module in Continuous Process Auditing (CPA) for real-time Decision Support systems [13]. Though chronological, topological, and other types of semantic relation already exist [74], in these methods only hierarchical concepts are extracted and reduced sets of semantic [14] relations are in use.

According to Wache [85], the hybrid approach to integrating ontologies is the most relevant approach because it allows for semantic heterogeneity and flexibility. Pervasive systems are complex and heterogeneous in nature and their data sources are semantically heterogeneous. A common ontology approach is straightforward for dealing with heterogeneous semantic data sources through hybrid approaches and multiple ontologies [46]. Integration and making mappings of ontologies are necessary for this kind of scenario. In this context, a common ontology has proposed to construct from multiple Process Ontologies (PO) with integration of semantically heterogeneous data sources. A Mapping Ontology (MO) allows mappings between the various PO and it also seeks and relates mappings between the various schema of

the integrated data. To represent the mappings automatically, or in a semi-automatic way, an expert system like JESS [31] is merged with the PO to infer new relations from the existing concepts.

The objective of this chapter and the next chapter is to present a hybrid ontology construction approach that combines the common ontology (shared vocabulary) with multiple, integrated and mapped POs. This common ontology would be the top-most layer in topological semantic relations. Audit Rule Ontologies for Processes (AROPs) would be stemmed as the second layer, under the common ontology layer in CPA, to infer the audit rules. The semi-automated development of the Common ontology in the context of healthcare based on the proposed approach is presented in next Chapter 7.

6.2 Knowledge Representation and Ontologies

As defined by Gruber, an ontology is an explicit specification of conceptualization [36], that can serve as an effective and powerful tool to capture, store and work with domain knowledge in knowledge-based information systems. In terms of knowledge representation, there are several types of ontology, including high-level, generic, domain and application. Generally speaking, domain ontologies are intended to specify the conceptualization of particular real-world domains. Domain ontologies usually describe a set of concepts and activities related to domains such as finance, commerce or industries involved in the production or delivery of goods and services, and other examples.

Below we describe the ontology aspects relevant to domain and processes, then audit rules and finally hybrid approaches in next chapter.

6.2.1 Common Ontology or Shared Vocabulary

Generally speaking, domain ontologies are intended to specify the conceptualization of particular real-world domains. Domain ontologies usually describe a set of concepts and activities related to domains such as finance, commerce or industries involved in the production or delivery of goods and services, and other examples. There are two different ways of representing knowledge through domain ontology: (a) single domain ontology that covers the complete domain, and (b) multiple sub-domain ontologies to cover the complete domain. Since later approach has the versatility of adding more than one sub-domains, sub-domain ontologies must be aligned first to construct the Common Ontology or Shared Vocabulary. This is done by defining Common Ontology (CO) and by defining mappings between the sub-domain ontologies and this Common Ontology (Figure 6.1 and 6.2). Then, the concepts used in the Process Ontology can be defined in term of the concepts of this Common Ontology, and the Process Ontology will be viewed on the Common Ontology.

6.2.2 Process Ontology

All activities in a process are linked as sequential steps either defined by higher business modelers or discovered by various established methodologies, such as workflow mining from labeled and unlabeled event logs, stochastic workflow analysis, rule-based approaches and so on. There are two system approaches to the study of any system and its behavior: the micro system (μ), studies the algorithms, sensors for collecting data, and atomic devices; and the macro system (M), which studies and models large systems composed

of large numbers of algorithms and devices. Fig. 6.3 depicts the pictorial view of a process traversal in the macro system (M). For example, Process P1 traversed through $\mu_1, \mu_5, \mu_2, \mu_4$ respectively then terminated at μ_3 in a macro system (M), which can be viewed as directed acyclic graph (DAG). Though, a DAG is directed graph with no directed cycles and it must not contain any cycles or loops between macro systems but a process may hold a cycle or loop within a macro system. In macro system viewpoint and in the context of Continuous Process Auditing (CPA), DAG is an appropriate way to view and represent a process. Discovering process in a macro system, a NP-hard problem, is similar to find a topological ordering of a process in a given DAG. Our focus is on already discovered or pre-defined; solving the problem of discovering processes in a macro system will be investigated in future research within the context of Ontology Evolution.

Process Ontologies (PO) are constructed for each process with their defined concepts and databases that might be either homogeneous or heterogeneous in nature, and an expert system for PO mappings is coupled to construct our proposed hybrid layered ontology approach addressed in more detail in next Chapter 7.

6.2.3 Process Ontology Example

A Dataflow Diagram (DFD), is intended to show the *functionality* of a system, consists of a collection of processes, storages, terminators linked by flows. Following DFD example is taken from 'The MIT Process Handbook' by Malone et al. [18] in Figure 6.4.

In the DFD diagram, Processes are shown as circles. Each process is the component actions or subprocesses which together constitute the overall process or system being represented in the diagram. The workflow of procedures "Cervical Malignancy" patient from Dutch Academic Hospital [75] who has taken services from both Diagnostic and Therapeutic Services as part of her treatment is presented below in Figure 6.5. Process Ontology of this workflow process is depicted in Figure 6.6.

6.3 Conclusion and Outcome

A Common Ontology serves the purpose of gathering domain knowledge of a set of concepts and activities related to the domain. Its multi-layered approach also represents knowledge through from a single complete domain to multiple sub-domain ontologies. Because of the versatility of adding more than one sub-domain, we are relying on the multiple sub-domain ontologies which must be aligned first to construct Common Ontology (or Shared Vocabulary). Process Ontology is the translation of the process workflow procedures that must be mapped on top Common Ontology for rule based accessing to distributed heterogeneous sources. Rule based accessibility and methodology are devised in the next Chapter 7, along with the mechanisms for defining Audit Rules.

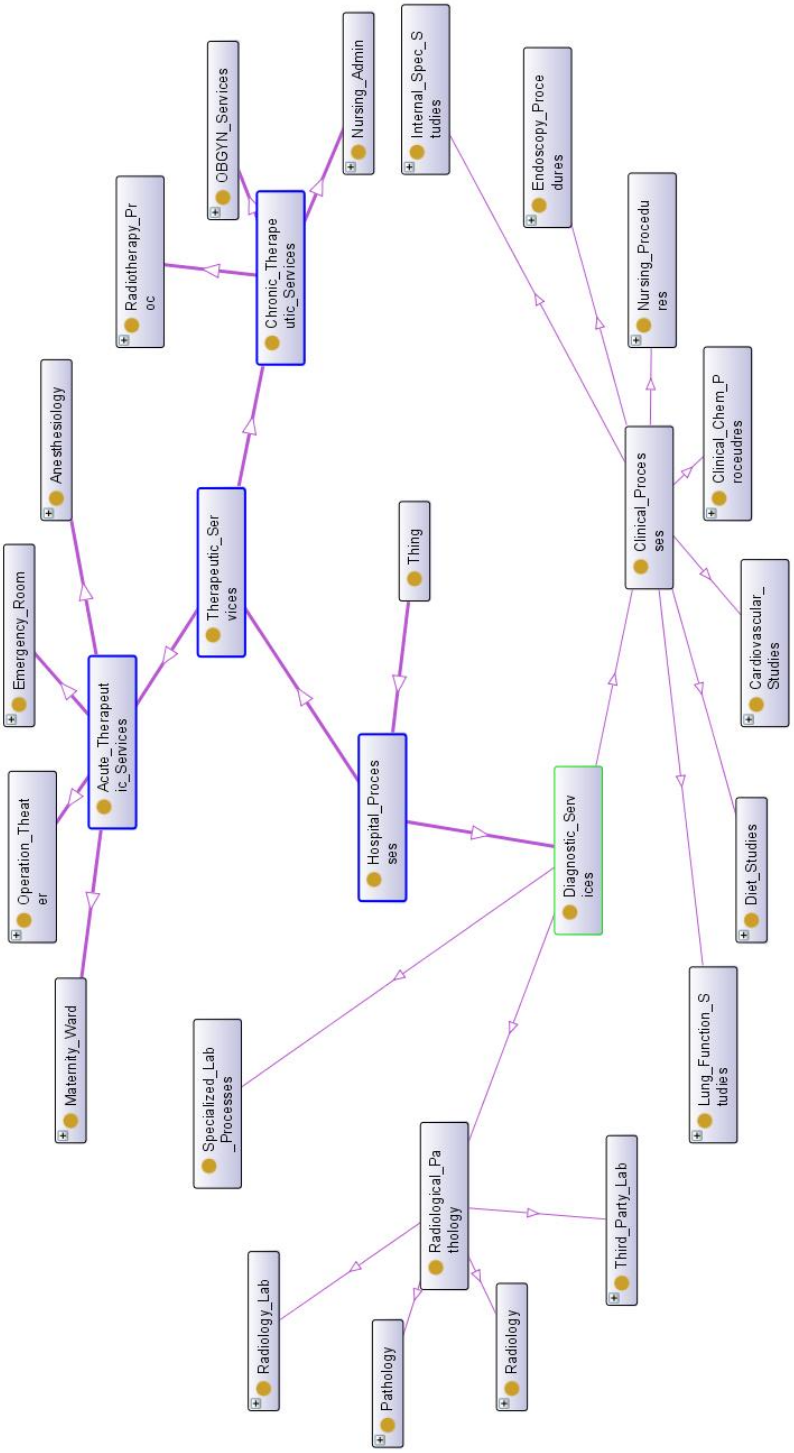


Figure 6.1: Common Ontology of Dutch Academic Hospital [75]

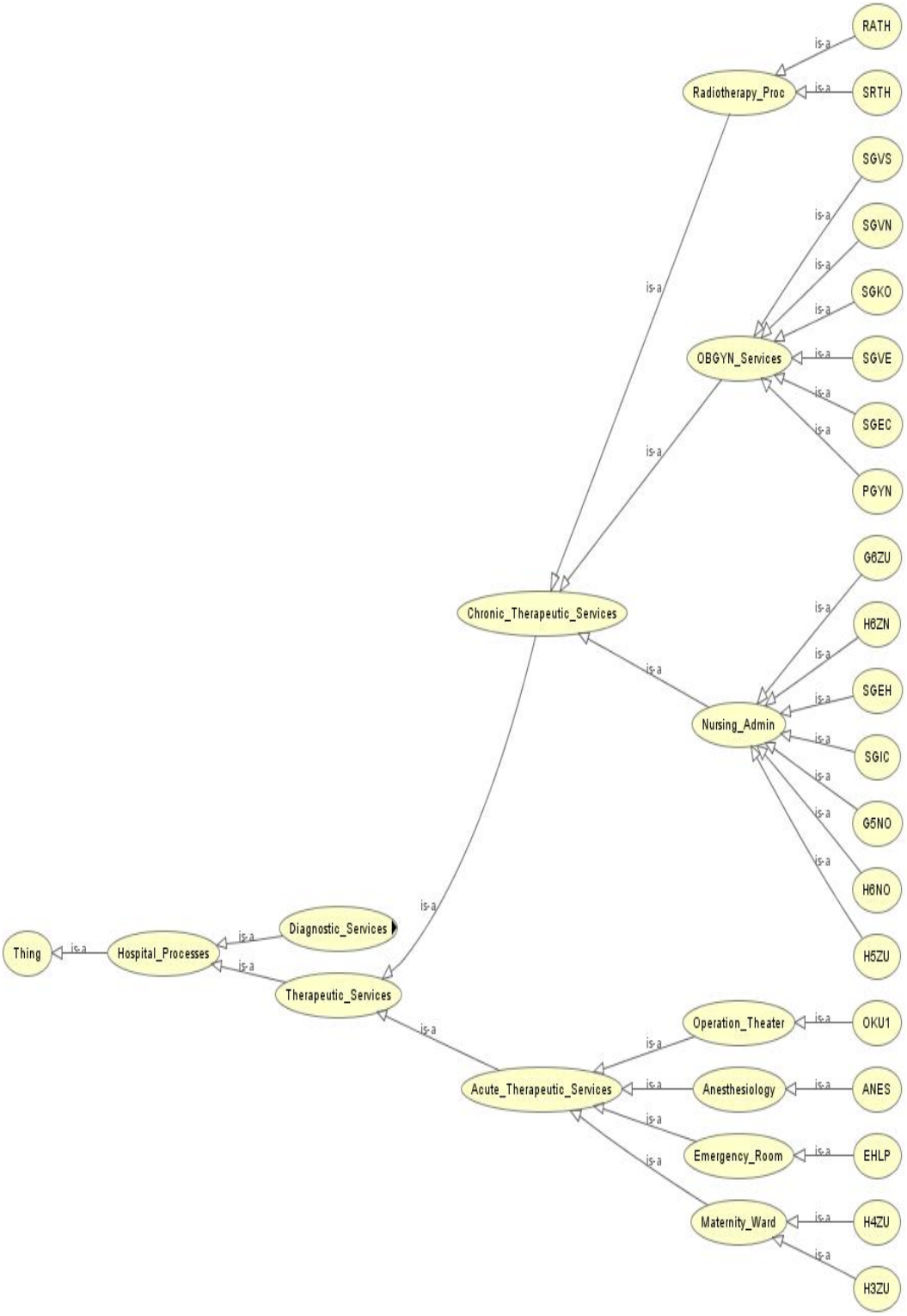


Figure 6.2: Common Ontology of Dutch Academic Hospital of Therapeutic Services

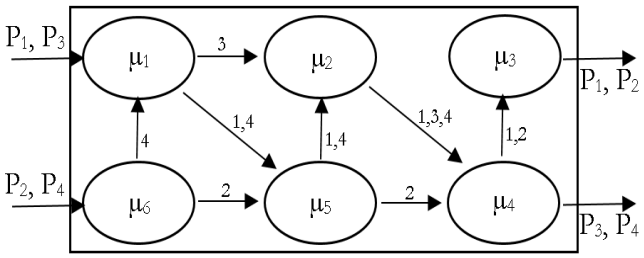


Figure 6.3: How a Process is traversing in a Macro System (M). For example, Process P2 traversed through μ_6, μ_5, μ_4 respectively then terminated at μ_3.

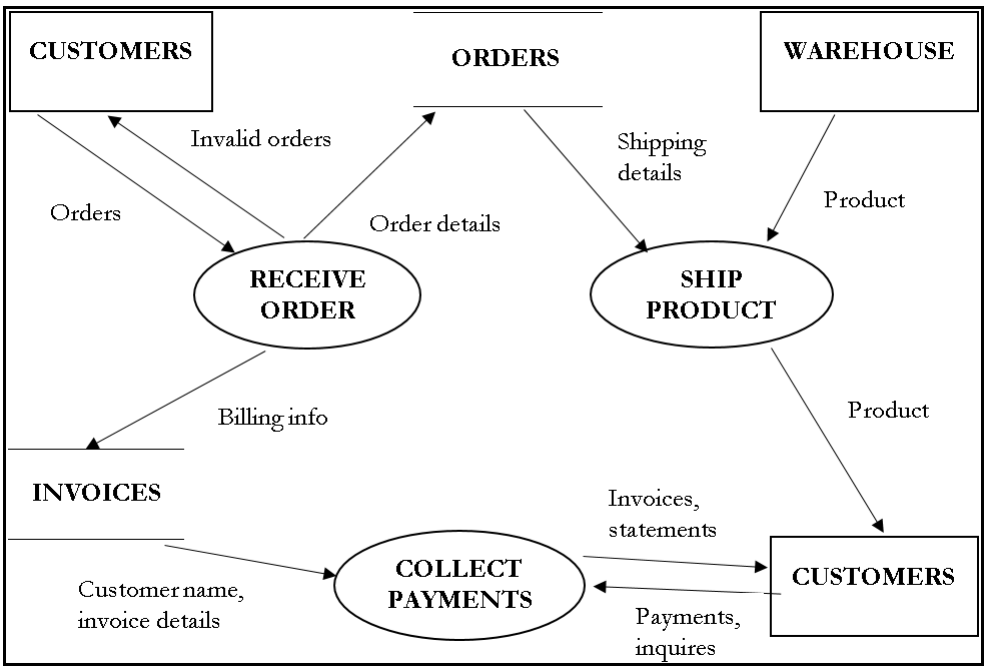


Figure 6.4: DFD of generalized form of Order Processing

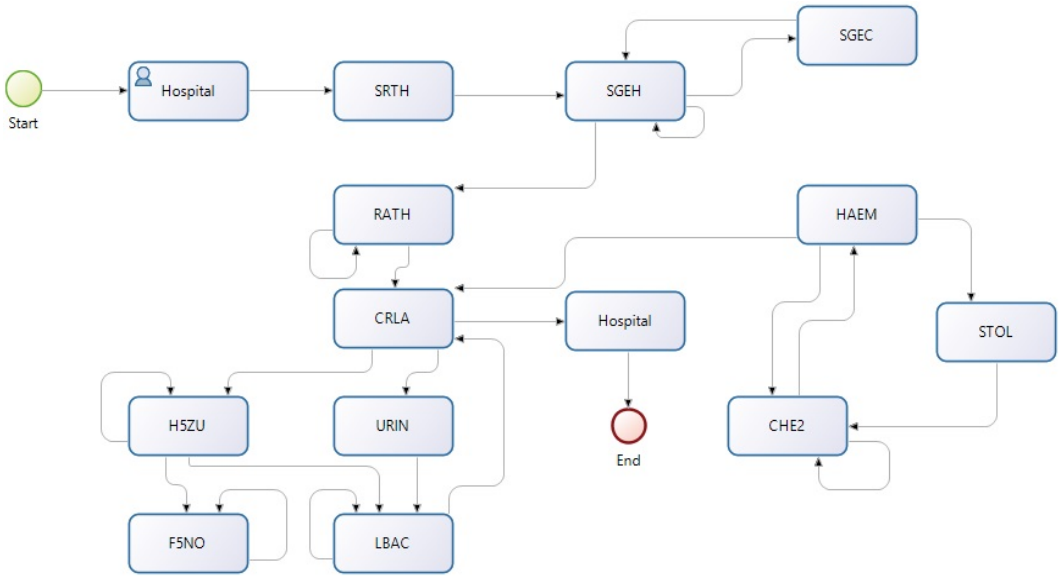


Figure 6.5: Workflow of procedures undergone by a "Cervical Malignancy" patient at the Dutch Academic Hospital

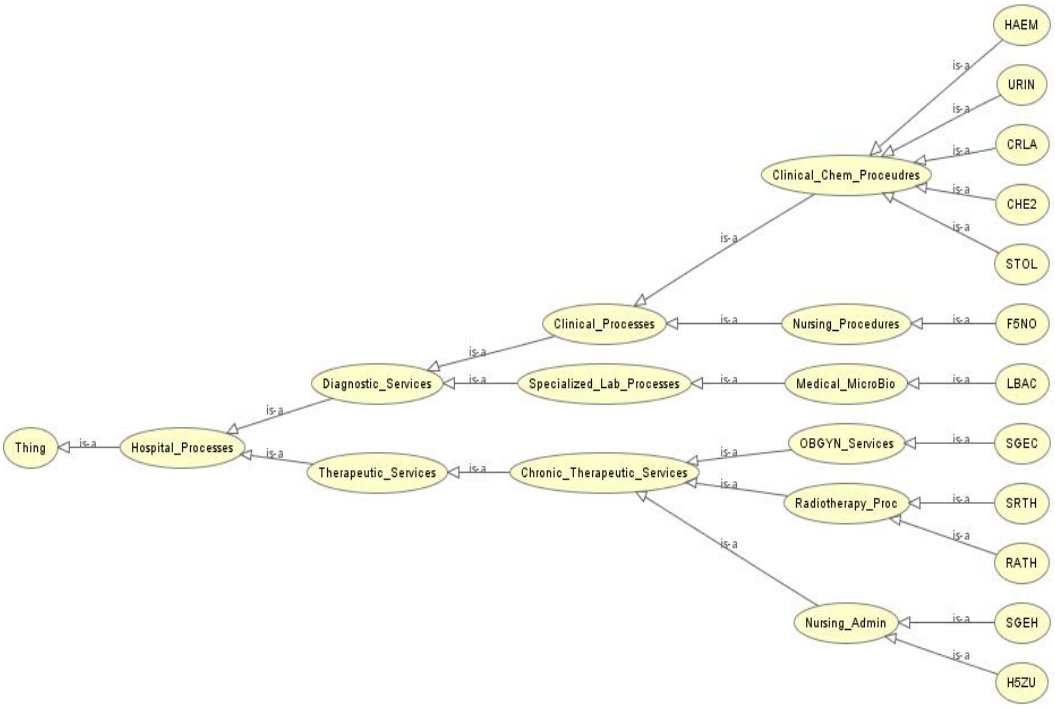


Figure 6.6: Process Ontology of a "Cervical Malignancy" patient at the Dutch Academic Hospital

Chapter 7

Audit Rule Ontology of a Process (AROP)

In Continuous Process Auditing, an audit rule sheet is defined by audit professionals for each process or module. The matter of continuity of application in CPA ranges from continuous time-dependent modeling to discrete time steps of audit application adapted to application requirements. For approaches with small time steps, the sheer magnitude of computational tasks to support CPA demands use of coarse-grained analysis of sub-systems, and estimation techniques based on limited rule sets. This consideration is used to determine the degree of conceptualization as knowledge, audit measures using sensors and reasoning through rules and inference.

Though some processes are co-related to each other, we assume all processes are independent of each other. Since our main research interest is on the continuous auditing of processes, all audit rules are defined for a spe-

cific independent process. Audit rules must take place with their precedent and hierarchical level of events captured through Process Ontology (PO). Our proposed Audit Rule Ontology of a Process (AROP) will be constructed by using audit rules for that process which maintains precedent and hierarchical occurrence of events.

To represent common knowledge and processes in Continuous Process Auditing, we have proposed our ontological approaches in Chapter 6 with the vision of developing rule-based Continuous Process Auditing (CPA) methodologies and a CPA system, ultimately. Defining audit rules and constructing their AROP is the vital components of a proposed model described in Section 7.4. A conceptual view of rule-based Audit Rule Ontology of a Process (AROP) in Hybrid Layered ontology model is depicted in Figure 7.1.

In this Chapter, we have presented a mechanism to construct audit rules that are defined by audit professionals and a mechanism to construct their AROPs. An example of the development cycle (from conceptualization to construction to operation) of Hybrid Layered Ontology is also presented in the context of Healthcare system. An approach to map between Domain and Process ontologies, and Audit Rule Ontology of a Process (AROP) is also described.

7.1 Audit Rules

Audit rules are defined for an activity or a system component within the audit scope in any human-performed auditing mechanism, as a first step of common auditing procedures. Audit rules are based on both the hierarchical

structure of organization and the enforced business controls. The same audit rules discernment and definition approach can be applied and implemented in any Continuous Process Auditing system where a process has to traverse through various components by applying audit rules sequentially.

7.2 Audit Rule Ontology of a Process (AROP)

An Audit Rule Ontology of a Process (AROP) would be used to detect exceptions to the audit rules in a process during CPA. Semantic rule-based reasoning would facilitate construction of Audit Rule Ontology of a Process (AROP) in a semi-automatic way. AROPs would be used as second layer under common ontology in hybrid layered ontology model. We assume that human approval of all audit rules is enforced; the issue of autonomous automated approval through artificial intelligence is not considered in this discussion.

7.2.1 Semantics of AROP

A rule is a proposition that is a claim of obligation or of necessity [34]. An audit rule is derived from audit policy that is a 'directly enforceable' element of governance and a proposition. Element of governance is concerned with directly controlling, influencing, or regulating the actions of an enterprise and the people in it. Violations of the element of governance can be detected without the need for additional interpretation of the element of governance. 'Directly enforceable' means that a person who knows about the element of governance could observe relevant business activity (including his or her own behavior) and decide directly whether or not the business was complying with

the element of governance. The audit rules that we use in practice mostly are of the following nature:

- *Structural or definitional audit rule* that is a claim of *necessity* and each structural audit is *practicable*. Structural necessity should be **verifiable** either directly or indirectly.

- *Operative or behavioral audit rule* that is a claim of *obligation*. This is an element of governance that is **directly enforceable**. Since, each element of governance that is directly enforceable is also *practicable*, no operative audit rule is a structural audit rule.

- *Authorization audit rule* that authorizes or forbids certain assignments or tasks to agents, modules or roles. This element of governance that is also **directly enforceable**.

The semantic relations among audit rules are mainly as follows:

- *Domain* is a class relation

- *Process* is sub-class of domain class

- *Rule-group* is a sub class of process class

- *Rule* is a sub-class of Rule-group class. Following are the generalized abstract form of few audit rules:

  - *Verify* how we think about things - what is meant by a word (a concept) or by a statement (a proposition) or by a question. See Appendix A for various meanings of concepts and propositions.

- *Strict* strictly enforced (If you violate the rule, you cannot escape the penalty.)
- *Deferred* deferred enforcement (Strictly enforced, but enforcement may be delayed e.g., waiting for resource with required skills.)
- *Pre-authorized* pre-authorized override (Enforced, but exceptions allowed, with prior approval for actors with before-the-fact override authorization.)
- *Post-justified* post-justified override (If not approved after the fact, you may be subject to sanction or other consequences).

7.2.2 Audit Rule: Use Case and RDF Triples

We define an audit rule and describe the entire step-by-step procedure as functional requirements (FR) specification for constructing an AROP. To illustrate this we consider an example from billing, namely:

- *FR 1*: Define the audit rule in natural language (i.e. in English) form? Verify ZIP code in customers billing address.
- *FR 2*: Break down the audit rule formable triple logic format and formalize it in Descriptive Logic? Billing address **isa** address; Street is **partOf** address; City is **subClassOf** State; State is **Class** of state code; ZIPcode is **partOf** address; ZIPcode **hasValue** 5 numeric digit; so on
- *FR 3*: Define the semantic schema using object(individual) properties

(IF, THEN, AND, OR, NOT, HasValue) and classes(concepts) (Variable, Value, Rule, and Rule-group)?

```
IF object = billing_address
AND "entered_ZIP" < 99999
AND "entered_ZIP" > 9999
AND "zip_range" = 99999:99999
THEN verified = TRUE
```

- *FR 4*: Construct RDF and RDFS triples?

```
<variable rdf:ID = entered_ZIP/>
<variable rdf:ID = object>
<HasValue rdf:resource = #billing_address/>
</variable>
<variable rdf:ID = zip_range>
<HasValue rdf:resource = #48150:49999/>
</variable>
<variable rdf:ID = verified>
<HasValue rdf:resource = #TRUE/>
</variable>

<Rule rdf:ID = ZIP_verify>
<IF rdf:resource = #entered_ZIP/>
<IF rdf:resource = #zip_range/>
<IF rdf:resource = #object/>
```

```
<THEN rdf:resource = #verified/>
</Rule>
```

7.3 Ontology Design: A Hybrid Layered Model

Domain ontologies may be divided into linguistic and conceptual ontologies. According to Gruber [36], Conceptual Ontologies (CO) represent the domain objects, distinguishing between the primitive concepts and the defined concepts, whereas Linguistic Ontologies (LO) define words or contextual usages of words. The Process Ontology (PO) contains only the defined concepts and the Mapping Ontology (MO) contains both the defined and the underlying primitive concepts. The observation in [42] led to identifying some relationships between POs, MOs and LOs. Mappings between POs may be defined in terms of equivalence operators of some MO. The various meanings of words in MO references may be defined by LOs and this reference would provide a basis for formal, and exact or uncertain reasoning, and automatic translation of context-specific terms.

7.4 Proposed Hybrid Layered Approach

We propose a single common ontology approach with multiple POs for domain ontology construction. Each PO is attached to a database that might be heterogeneous in nature with other databases. Each PO describes the semantics of data sources individually. Inter-PO mapping is realized by the MO, which is defined with primitive concepts. The simplicity and flexibility

permitting addition of new sources (like new POs) with little or no need of modification, is the main advantage of this mechanism. To integrate several POs addressing the same domain, this mechanism exploits the MO's capability to define equivalent and similar concepts. We discuss three mapping use cases of semantic integration:

Discovery of Mapping: To find the similarities and determine the concepts and properties for representing similar notions between two POs, we use PO structures, definitions of concepts, and instances of classes.

Mappings Representation of MO: To represent the mappings between two POs to enable reasoning with mappings. The mappings representation of inter-PO is used in defining the MO. The MO can consist of the OWL constructor or the equivalence relation defined, or we can use different inference engines to assert the MO automatically, starting from the logical rules between the concepts belonging to different PO.

Mapping Uses: How to define the mapping between PO is not described or not a goal in itself, but it can be done either automatically or interactively. The resulting mappings are used for various integration purposes, such ase answering the users requests. We use JESS instructions language for this purpose.

Using an example case from healthcare, as detailed by Wache [85], the first step is to develop the Healthcare Common Ontology using multiple POs in the top-most layer, then Audit Rule Ontologies of Processes would be stemmed as a second layer under the top-most layer to form a Hybrid Layered Ontology.

Figure 7.1 visualizes the conceptual model of both the layers alongside with the technologies would be used to integrate and to develop the whole operable Audit Rule Ontology system [71]. More abstract description of construction, development and operational mechanism are discussed in the next section. Figures 7.2 and 7.3 depict the abstract relational view and semantic view of AROP, respectively. A semantic view of "Verify" audit rule, as an example, of Healthcare diagnostic process is presented in Figure 7.4.

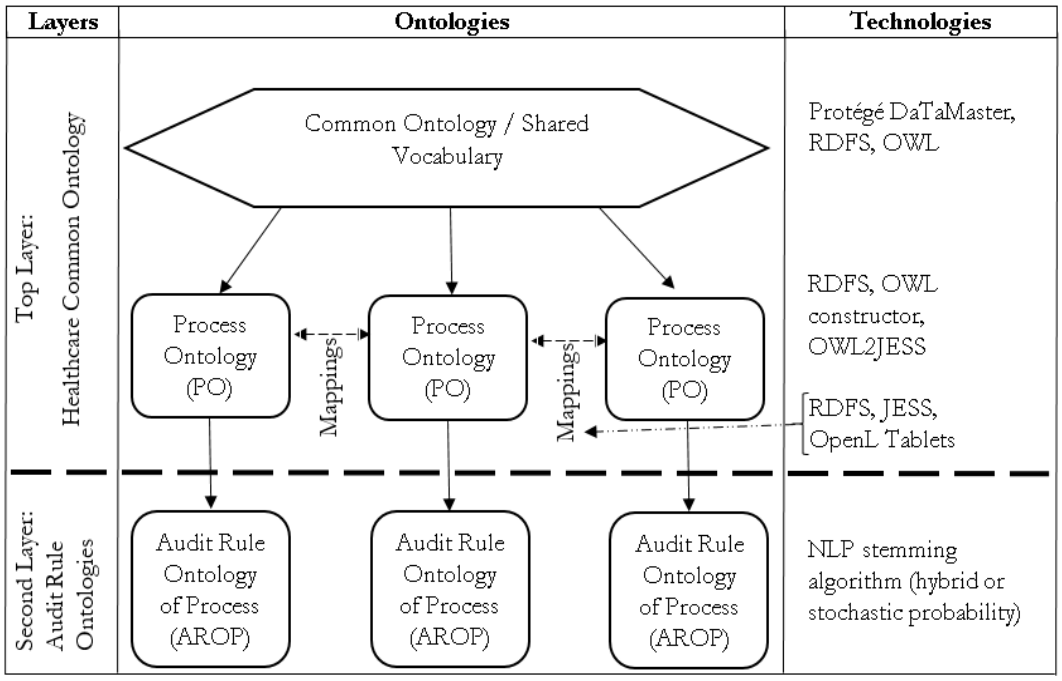


Figure 7.1: Conceptual Model of Hybrid Audit Rule Ontology [71].

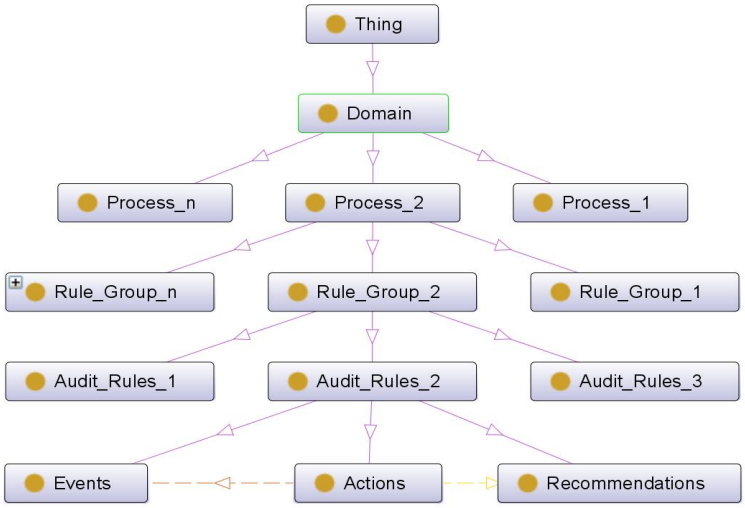


Figure 7.2: Abstract Relational view of AROP

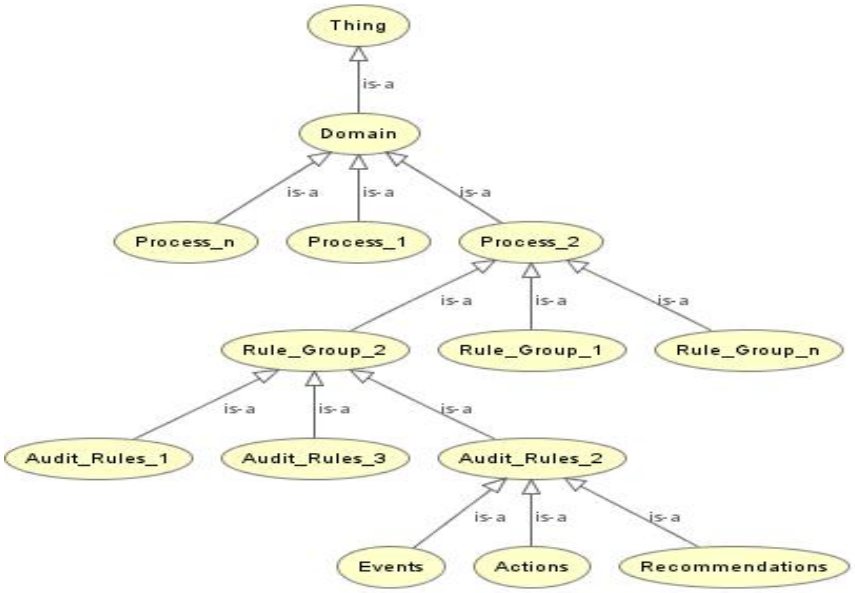


Figure 7.3: Abstract Semantic view of AROP

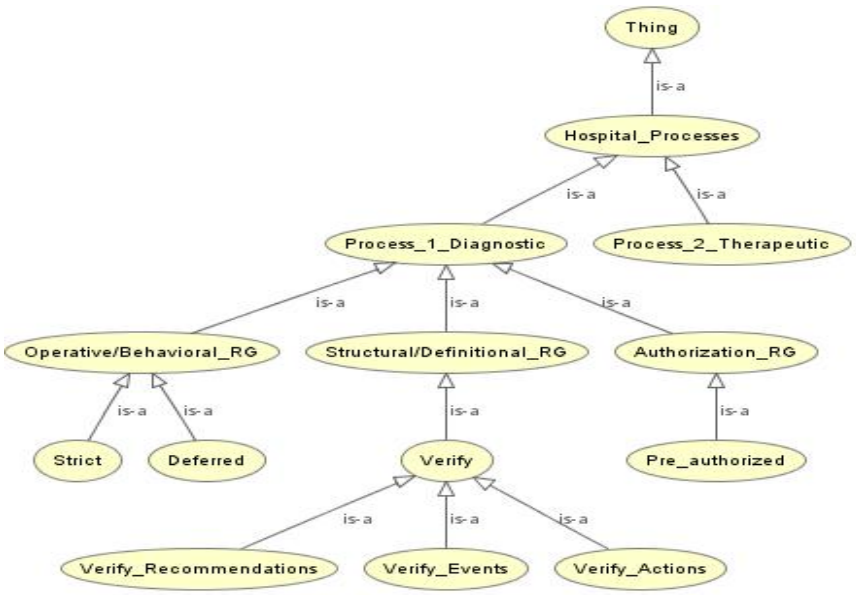


Figure 7.4: Semantic view AROP example with "Verify" Audit rule expanded

7.5 Development of Healthcare Common Ontology

Many business changes have occurred throughout history and a recent major change is related to Enterprise Information Systems (EIS) and the methodology Enterprise Resource Planning (ERP) [43]. These approaches have transformed the way business data is collected, stored, disseminated and used [73]. Enterprise resource planning systems are defined as "information system packages that integrate information and information-based processes within and across functional areas in an organization" [50]. An ERP system is an enterprise system that affects many or all departments of a company. Though research of Continuous Auditing started in 1991 [80, 24, 81]; Kent et. al. [45] first envisioned the application of Continuous Auditing for Healthcare

Decision Support Systems in 2010.

Ontologies have been used to represent knowledge and to help knowledge inference in clinical research [3], decision support and maintenance of system [64]. Jean et al [42] showed the specification of an ontology as a domain model allowing solutions for various issues in data indexing, data exchange and data integration. An ontological representation with rule-based reasoning as model for development of clinical decision support system was presented by Archour et al [3]. Alles et al discuss Continuous *Process* Auditing (CPA) in enterprise system environments related to Healthcare Decision Support System, requires definition and discovery of processes [4, 10]. Each process can be constructed as a Process Ontology (PO), as described in the previous section.

Each data source (DS) containing the process knowledge or information is described by a PO. For example, PO-DS1 and PO-DS2 for the first and second data source, respectively, with the ability to obtain access to identifiable POs. All available data sources are merged to construct a common domain ontology. The common vocabulary of each data source becomes a sub-group of the common domain ontology.

The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. Shared data representations such as eXtensible Markup Language (XML) provides an elemental syntax for content structure within documents lies in the bottom of the Semantic Web Stack. Resource Description Framework (acrshortrdf) and RDF Schema (RDFS), a general method for describing information are on top of XML in the stack respectively. Web Ontology Language (OWL) [53], a family of knowledge representation languages lies on top of

RDFS in the same semantic web stack, adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. OWL provides the tools for semantic reasoning to describe or to represent knowledge. The OWL constructors, and the equivalent relation (such as Rule Interchange Format (RIF)) or the specific relation of the domain to be created between concepts belonging to different POs would be the MO operators.

7.5.1 Healthcare Common Ontology Conceptualization

The Protégé Plug-in DaTaMaster [58] is used to implement the proposed solution. Protégé Plug-in DaTaMaster imports data source schemes and their contents under OWL. It permits the integration of various data sources in single ontological representation. To implement the solution a) any data connection driver like ODBC or JDBC can be used to connect with data bases b) selection and visualization of table content by the user preference, and c) each activated and visualized table is transferred into a class or sub-class depending on the choice of user.

7.5.2 Healthcare Common Ontology Construction

The insertion of OWL constructors, relations as well as annotations participate in the process of semantically enhance data belong to different PO as well as to solves syntaxes and semantic heterogeneity of integrated systems, improving data exchange between them. We assigned a unique space name to

each PO to resolve conflict context. A pre-tagging concept is used to pre-tag with the same Uniform Resource Identifier (URI) for all the classes, attributes and the instances belonging to one PO. We create relations to resolve the naming conflict. Two classes issued from two different data sources (different PO), where first one describes the designation or equipment (id, MRI machine, ER room, pharmacy, diagnostic result) and the second details a patient cases (id, symptoms, current status, drugs). Both classes treat the same patient or equipment but the semantic of their data sources are different. A manual relation was created between the both classes that maintains an equivalent relation for both instances. For large ontology, automatic definition of the MO can be implemented with JESS [31] instructions to discover the common attribute.

7.5.3 Healthcare Common Ontology Operation

A rule based reasoning engine like JESS [31] can be used with the ontology instances. Tools like OWL2JESS [54] and OpenL Tablets [61] facilitate the necessary conversion of the OWL ontology code to facts and rules. RDFS and OWL verified coherence and uniformity of the ontology will permit us to design, evaluate and refine the original obtained ontology. Three layers of knowledge: the ontology model layer, the ontology layer and instances layer encapsulated by the obtained knowledge base.

OpenL Tablets is an open source rules engine and rules repository tool [61] which has Java Wrappers that can integrate and enable interoperability with JESS rules and instructions. The exploitation of the ontology is ensured

using rules and requests permitting fetching from the knowledge base through a set of JESS commands, such as “defrule” and “defquery”

7.6 Mappings Between Domain and Process Ontology

Mapping ontology between domain, process and AROP is depicted in Figure 7.5. Many process ontologies are aligned by defining a domain ontology mapping between the process ontologies and the domain ontology. Additional concepts which not present in the domain ontology can be defined using this domain ontology mapping as well. The AROP reference mapping defines the references of AROPs on the process ontologies because the conceptualization of domain ontology may not always exactly suit the requirements of AROPs [23].

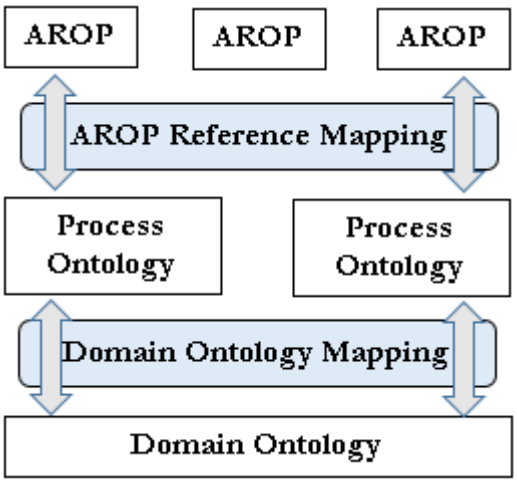


Figure 7.5: Mapping between ontologies (domain, process and AROP)

7.7 Conclusion and Outcome

A domain ontology such as Hybrid Layered Ontology is an approach that permits very efficient knowledge management and gives a unified conceptualization of the domain. In this chapter an approach using multiple Process Ontologies based on knowledge extraction and integration from multiple, different data sources, which enhances semantically the final result, was presented. The semi-automatic ontology construction not only allows a faster and more efficient construction, but also may aid significantly in saving manual time consumption, effort and consistency. In order to define the Mapping Ontology automatically from the logical rules, an expert system JESS is integrated with OpenL Tablets. To update the knowledge base for better diagnoses and maintenance, a strategy for ontology evolution at conceptual, relational and instance levels are presented.

Part III

Evaluation and Applications

Chapter 8

Evolution of Ontology and Methodologies

Constant changes of business activities requires the change of business rules as well as audit rules for Continuous Process Auditing (CPA). The evolutionary process, a crucial part of the ontology lifecycle, creates newer versions with added stemming down the tree from the original ontology. Since the uniformity and coherence of the ontology must be respected, the evolution process is difficult to implement semi-automatically and should be considered as beyond human capacities for complex ontologies.

In this chapter, we have presented an algorithmic approach to identify either conceptual or application types of changes and to edit the identified changes in three ways to linking the conceptual and semantic relations. The coherence and uniformity of the ontology were verified by the predefined semantic Audit rules (in RDF) and OWL. Evaluation and refinement of the original ontology are permissible through coherence and uniformity. All three

layers of ontologies - Common Ontology, Process Ontology, and AROP were encapsulated by the obtained knowledge base that described in ontology operation in Section 7.5.

The proposed hybrid layered ontology, and development and operation of Common Ontology for Healthcare enterprises were presented in Section 7.4 and 7.5, respectively. We have carried the same Healthcare Common Ontology example to present our evolution methodology approach.

8.1 Ontology Evolution

Ontology evolution is defined as the process of updating the previous ontology version, in order to take into consideration changes within the domain, reflected in its conceptualization or application [21]. The evolutionary process, a crucial part of the ontology lifecycle, creates newer versions with added stemming down the tree from the original ontology. Since the uniformity and coherence of the ontology must be respected, the evolution process is difficult to implement semi-automatically and should be considered as beyond human capacities for complex ontologies

8.1.1 Evolution Methodology

Changes Identification of the domain is the fundamental step to form an ontology evolution strategy. There are two main methods to identify the changes - descending and ascending. The definition of domain update or the update of the ontology usage methods would be the *descending identification.* The

changes that identified from the ontology analysis itself, i.e. by using heuristic rules or statistical inference for optimization, are the *ascending identification*.

Editing ontology changes might be either elementary, intermediate or complex. Stanjanovic discussed the *elementary changes* as non-decomposable changes given by a suppression or adding of ontological entities [69]. *Complex changes* are composed of two or more elementary changes forms that together form a logical entity. Giorgos et. al. [32] described the complex changes may be composed of other types of changes that happen in between elementary and complex, and which may be called *intermediate changes*. The insertion of links belongings to the concept along with other existing concepts must be done after the addition of a new concept. The isolation of one or more concepts may be the reason for the concept suppression. All conceptual or semantic relations, linking the suppressed concept with other concepts must then be deleted along with linked instances.

8.1.2 Healthcare Common Ontology Evolution

The evolution of the addition, suppression and modification operations on the knowledge base using the JESS language "assert" for addition, "modify" for modification, "retract" for suppression. Mei et al. [54] showed how to transform the OWL to JESS facts. The JESS facts are of triplets type given by (Predicate, Subject, Object). It is compulsory that the development of a system which propagates the changes automatically. In order to make changes transparent, the system must guide the user during operation. The change may be implemented either on the ontological level or on the instances level.

Adding, suppressing and changing a concept, a relation or a semantic relation and updating conceptual relations may implemented on the ontological level whereas adding, suppressing and changing an instance should be implemented on the instances level.

An ontology evolution repository may be maintained for previous version and the process the changes made to the ontology. The repository would preserve the history of different versions of the ontology which helps to make changes forward and backward compatible.

8.2 Conclusion and Outcome

A domain ontology such as Hybrid Layered Ontology is an approach that permits very efficient knowledge management and gives a unified conceptualization of the domain. In this chapter an approach using multiple Process Ontologies based on knowledge extraction and integration from multiple, different data sources, which enhances semantically the final result, was presented. The semi-automatic ontology construction not only allows a faster and more efficient construction, but also may aid significantly in saving manual time consumption and effort, and improve accuracy and consistency. In order to define the Mapping Ontology automatically from the logical rules, an expert system JESS is integrated with OpenL Tablets. To update the knowledge base for better diagnoses and maintenance, a strategy for ontology evolution at conceptual, relational and instance levels are presented.

Chapter 9

Evaluation of CPA Methodologies

9.1 Introduction

In this chapter we present our approach to evaluation of our CPA methodologies in two categories: (i) accessibility of distributed heterogeneous data sources, and (ii) Audit rules and Continuous Process Auditing. Accessing the distributed data sources in a short period of time, or real-time, is a complex problem. Our proposed Common Ontology and Process Ontology used to solve this problem within distributed environments for CPA is evaluated in Section 9.3.

In Section 9.4, our proposed Rule-based CPA methodology using Audit Rule Ontology of a Process (AROP) along with Common Ontology and Process Ontology in hybrid layered ontology is evaluated to find the answers to how audit rules are defined and used for detecting evidence and for control monitoring. We have devised ways to experiment with these methodologies using three distinct datasets from three different pervasive environments, in-

cluding the continuous assurance and monitoring of healthcare decision support [75], e-commerce [76], and production system [70] processes.

9.2 Description of Datasets

We have exhaustively searched to get a live or simulated systems to test our hypotheses and methodologies within both for-profit and non-profit organizations. We had even discussion regarding the research collaboration with well organizations like KPMG, Health Canada and Henry Ford Hospital but we are yet to find a research collaboration partner that willingly to create shareable environments for our novel research in Continuous Process Auditing. Jans, Alles and Vasarhelyi discussed the Process Mining of Event Logs in Auditing in [40, 41]. They have presented their findings on the opportunities, challenges and areas of application for Process Mining of Event Logs in Auditing. This actually helped us to find the following datasets that contain the real-life captured event logs in three pervasive environments [75, 76, 70].

9.2.1 Healthcare Decision Support System (HDSS)

This dataset contains real-life log captured in a Dutch Academic Hospital [75]. This log contains some 150,291 events in over 1143 processes (cases). Apart from some anonymization, the log contains all data as it came from the Hospital's systems. Each case is a patient of a Gynaecology department. The log contains information about when certain activities took place, which group performed the activity and so on. Many attributes have been recorded that are relevant to the process. Some attributes are repeated more than

once for a patient, indicating that this patient went through different (maybe overlapping) phases, where a phase consists of the combination Diagnosis & Treatment.

9.2.2 E-commerce Management System (EMS)

This is a real-life log [76], taken from a Dutch Financial Institute. This log contains some 262,200 events in 13,087 processes (cases). Apart from some anonymization, the log contains all data as it came from the financial institute. The process represented in the event log is an application process for a personal loan or overdraft within a global financing organization. The amount requested by the customer is indicated in the case attribute `AMOUNT_REQ`, which is global, i.e. every case contains this attribute. The event log is a merger of three intertwined sub processes. The first letter of each task name identifies from which sub process (source) it originated from. A process tree of Dutch Financial Institute's personal loan application process is depicted in Figure 9.1 and maps of the same process and sub-processes is depicted Figure 9.2.

Processes (cases): **13087**, Events captured: **262200**, Event classes: **36**. Table 9.1 and 9.2 showed the absolute and relative occurrences of all events captured, start and end events.

9.2.3 Production Management System (PMS)

This is an event log [70] from Volvo IT Belgium. The log contains events from an incident and problem management system called VINST. The *incident management* process aims to restore normal service operations after

All event classes (36)		
Event classes	Occurrences (absolute)	Occurrences (relative)%
W_Completeren aanvraag+COMPLETE	23967	0.09141
W_Completeren aanvraag+START	23512	0.08967
W_Nabellen offertes+COMPLETE	22976	0.08763
W_Nabellen offertes+START	22406	0.08545
A_SUBMITTED+COMPLETE	13087	0.04991
A_PARTLYSUBMITTED+COMPLETE	13087	0.04991
W_Nabellen incomplete dossiers+COMPLETE	11407	0.0435
W_Nabellen incomplete dossiers+START	11400	0.04348
W_Valideren aanvraag+COMPLETE	7895	0.03011
W_Valideren aanvraag+START	7891	0.0301
A_DECLINED+COMPLETE	7635	0.02912
W_Completeren aanvraag+SCHEDULE	7371	0.02811
A_PREACCEPTED+COMPLETE	7367	0.0281
O_SELECTED+COMPLETE	7030	0.02681
O_CREATED+COMPLETE	7030	0.02681
O_SENT+COMPLETE	7030	0.02681
W_Nabellen offertes+SCHEDULE	6634	0.0253
W_Afhandelen leads+COMPLETE	5898	0.02249
W_Afhandelen leads+START	5897	0.02249
A_ACCEPTED+COMPLETE	5113	0.0195
W_Valideren aanvraag+SCHEDULE	5023	0.01916
A_FINALIZED+COMPLETE	5015	0.01913
W_Afhandelen leads+SCHEDULE	4771	0.0182
O_CANCELLED+COMPLETE	3655	0.01394
O_SENT_BACK+COMPLETE	3454	0.01317
A_CANCELLED+COMPLETE	2807	0.01071
W_Nabellen incomplete dossiers+SCHEDULE	2383	0.00909
A_APPROVED+COMPLETE	2246	0.00857
A_REGISTERED+COMPLETE	2246	0.00857
A_ACTIVATED+COMPLETE	2246	0.00857
O_ACCEPTED+COMPLETE	2243	0.00855
O_DECLINED+COMPLETE	802	0.00306
W_Beoordelen fraude+START	270	0.00103
W_Beoordelen fraude+COMPLETE	270	0.00103
W_Beoordelen fraude+SCHEDULE	124	0.00047
W_Wijzigen contractgegevens+SCHEDULE	12	0.005

Table 9.1: Dutch Financial Institute - All events

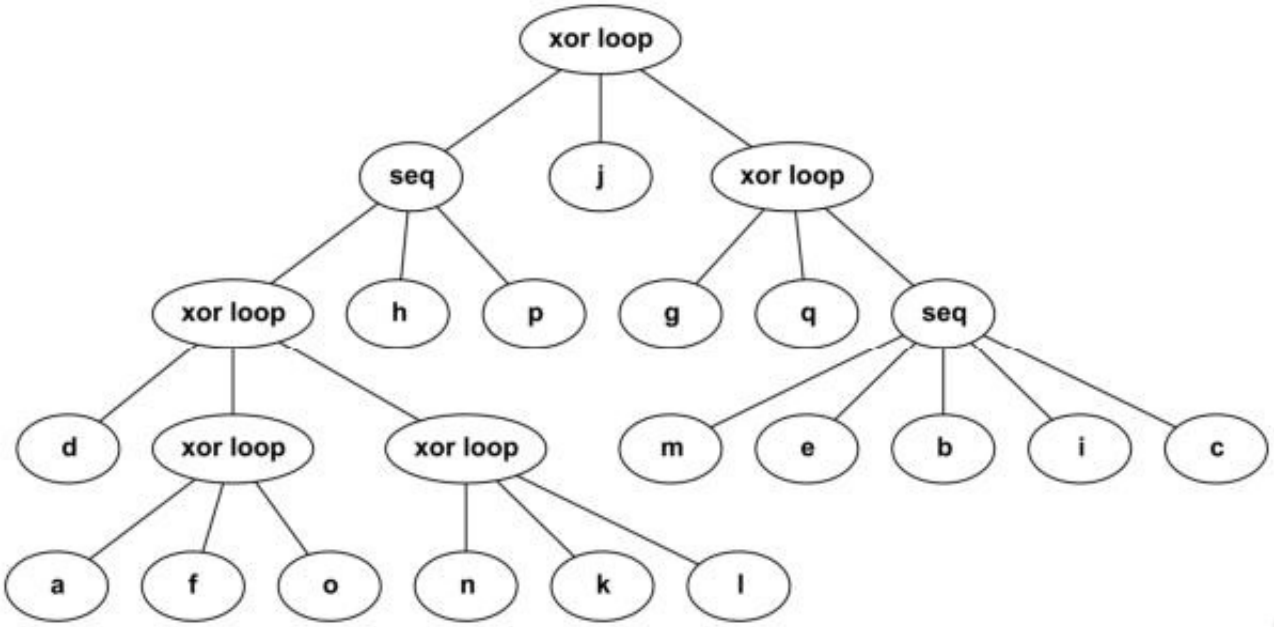


Figure 9.1: Dutch Financial Institute [76] - personal loan application *Process Tree*

the occurrence of a specific incident within SLA defined boundaries. Incident cases are first handled by a "first line" desk (the service desk and help desks) and escalated to second line and third line teams when the first line workers are not able to resolve the incident. Incident cases are assigned a priority level, which is calculated based on the impact (major, high, medium or low) and urgency (high, medium, low) of the issue, see Table 9.3.

Priority levels for incident processes are calculated based on the impact and urgency of the incident. When incidents are not resolved within a specified time frame, urgency is increased automatically. Incidents cannot automatically migrate from one impact level to another.

The *problem management* process tries to uncover the root causes behind incidents and implement fixes to prevent the occurrence of further incidents

Start event class (1)		
Captured events	Occurrences (absolute)	Occurrences (relative)%
A_SUBMITTED+COMPLETE	13087	100.0
End event classes (13)		
A_DECLINED+COMPLETE	3429	0.26202
W_Valideren aanvraag+COMPLETE	2745	0.20975
W_Afhandelen leads+COMPLETE	2234	0.1707
W_Completeren aanvraag+COMPLETE	1939	0.14816
W_Nabellen offertes+COMPLETE	1289	0.09849
A_CANCELLED+COMPLETE	655	0.05005
W_Nabellen incomplete dossiers+COMPLETE	452	0.03454
O_CANCELLED+COMPLETE	279	0.02132
W_Beoordelen fraude+COMPLETE	57	0.00436
W_Wijzigen contractgegevens+SCHEDULE	4	0.00031
W_Valideren aanvraag+START	2	0.00015
W_Nabellen offertes+START	1	0.00008
A_REGISTERED+COMPLETE	1	0.00008

Table 9.2: Dutch Financial Institute - Start and End events

—	Major Impact	High Impact	Medium Impact	Low Impact
High Urgency—	(1)	4	7	10
Medium Urgency—	(2)	5	8	11
Low Urgency—	(3)	6	9	12

Table 9.3: Impact and urgency levels of the incident management processes.

in IT-services operated by Volvo IT and includes activities to update internal knowledge bases with discovered findings. This process is primarily handled by second and third line teams. Contrary to the incident management process, there is no "push to front" system implemented for the problem management process which escalates cases among different service lines. The problem management and incident management processes thus work in parallel, where incidents are resolved as quickly as possibly in a "reactive" manner while the

underlying root cause is fixed by a problem management case.

The data set lists the following attributes for each logged event line:

"**SR Number**" (or "**Problem Number**" for the problem management process): the service request case identifier.

Example values: 1-364285768, 1-109135791, 1-147898401.

"**Change Date+Time**": the time stamp of the logged event line.

Example values: 2011-08-29T07:12:35+01:00, 2011-08-29T07:13:48+01:00.

"**Status**" and "**Sub Status**": the current status of the case as changed by the logged event line.

Example values: Queued/Awaiting Assignment, Accepted/In Progress, Accepted/Assigned, Closed/Cancelled, Unmatched/Wait

"**Impact**": level of impact the problem creates for the customer.

Example values: Major, Medium, Low, High.

"**Product**": the product involved in the case.

Example values: PROD821, PROD236, PROD793.

"**Involved ST**": the Support Team trying to solve the problem.

Example values: V5 3rd, V30, V13 2nd 3rd.

"**Involved ST Functional Division**": the support team's functional division.

Example values: V3_2, C_6, E_10.

"**Involved Organization**": the involved organisation line.
Example values: Org line A2, Org line C, Org line V7n.

"**Organization Country**": the location that takes the ownership of the support team.
Example values: fr, se, nl.

"**Owner Country**" and "**Owner First Name**": the person in the support team working on the case.
Example values: France/Frederic, Sweden/Adam, Belgium/Bert.

There are three types of logs captured from VINST: (i) incident management system log contains some 65533 events in over 7554 processes (traces), (ii) problem management system (open problems) log contains 2351 events in over 819 processes (traces), and (iii) problem management (closed problems) log contains some 6660 events in over 1487 processes (traces).

9.3 Accessibility of Distributed Heterogeneous Data Sources

The reality of business today is that no organization has data in a single place. Exceptions to heterogeneous environments and data sources are hard to find.

We need the ability to access data wherever it lives to get the answers we need. Make sure Continuous Control Monitoring and Auditing services in a process can access and handle data, everywhere in real-time. To evaluate the power of Process Ontology and AROP, we use the following database access protocols: ODBC, JDBC, OData, and MySQL.

From starting to complete end, processes are traversed through several layers of different departments and working units. For each dataset, we have physically set up the departments in two locations using 2 (two) servers with the capabilities of 10 (ten) virtually connected users for each servers. Event classes and real-life captured events were extracted from the datasets then loaded into the servers. Events were loaded accordingly with proper time stamp that need to be triggered.

All experiments described in section 9.4 of this chapter were conducted and data were collected for experimental results and analysis. We have experimented with a total of 20602 processes (healthcare 1143, e-commerce 13087, and production 6372) from each of the 03 (three) datasets for all of the above database access protocols. We have randomly sent the database access requests to all four database protocols from each processes. Access requests to database were sent by Audit rules and their AROP at the time of events triggering either to inferring rules or to enforcing actions.

9.4 Audit Rules and Continuous Process Auditing

9.4.1 Hypotheses and Design of Experiments

CPA methodologies such as Common Ontology, Process Ontology and Audit Rule Ontology of a Process (AROP) are designed to experiment using three separate datasets from three different pervasive environments. For each datasets, described in Section 9.2, we have designed and developed the Common Ontology for each domains, Process Ontology for each processes and AROP using all audit rules for a process that is mapped to a specific process ontology.

Healthcare Decision Support System Processes and Ontologies

We have developed the Common Ontology for Dutch Academic Hospital's Gynaecology department then we have chosen 4 processes to test our hypotheses of Process Ontology and AROP. Process Ontology (PO) have developed for each processes individually then hybridized with Common Ontology of Gynaecology department. Audit rules are constructed for each of the processes then AROPs have developed which are mapped to respective POs. To test all of these ontologies and to enforce the actions through audit rules, we have virtually reconstructed the whole Gynaecology department and all 4 processes within the department using events that are part of specific process. Sequential time-stamps also have carefully integrated in the process to preserve the process's sequential accessibility and accountability.

E-commerce System Processes and Ontologies

We have developed the Common Ontology for Dutch Financial Institute's Loan and Overdraft approval department then we have chosen 3 sub-processes to test our hypotheses of Process Ontology and AROP. The **A** subprocess is concerned with handling the applications themselves. The **O** subprocess handles offers send to customers for certain applications. The **W** process describes how work items, belonging to the application, are processed. Process Ontology (PO) have developed for each processes individually then hybridized with Common Ontology of Loan and Overdraft approval department. Audit rules are constructed for each of the processes then AROPs have developed which are mapped to respective POs. To test all of these ontologies and to enforce the actions through audit rules, we have virtually reconstructed the whole Loan and Overdraft approval department and all 3 processes within the department using events that are part of specific process. Sequential time-stamps also have carefully integrated in the process to preserve the process's sequential accessibility and accountability.

Production Monitoring System Processes and Ontologies

We have developed the Common Ontology for incident and problem handling system of Volvo IT which is called VINST then we have chosen 2 processes to test our hypotheses of Process Ontology, AROP of control monitoring rules. Handle Incidents process to ensuring the best possible levels of service quality and availability are maintained, and Handle Activity Problem process diagnoses the root cause(s) incidents activities and secures the resolution of

those problems to enhance the quality of IT-services delivered and/or operated by Volvo IT. Process Ontology (PO) have developed for each processes individually then hybridized with Common Ontology of incident and problem handling system. Audit rules are constructed for each of the processes then AROPs have developed which are mapped to respective POs. To test all of these ontologies and to enforce the actions through audit rules, we have virtually reconstructed the whole incident and problem handling system and all 2 processes within the department using events that are part of specific process. Sequential time-stamps also have carefully integrated in the process to preserve the process's sequential accessibility and accountability.

9.4.2 Experimental Approach and Settings

Healthcare Decision Support System Processes

HDSS dataset - an anonymized event log of a Dutch Academic Hospital [75]. Each process (case) is a patient of a Gynaecology department. The event log captures treatment procedures pertaining to 11 different diagnosis codes described in Appendix C. Various diversity of process instances were in the dataset in the event log with data attributes related to diagnosis and treatment. Event logs were filtered based on the properties such as diagnosis code, organizational data, time-sensitivity, treatment code, trace length, urgency, and specialism. Filtered event logs were also split into smaller and more homogeneous form of logs (e.g. patients having a particular type of cancer that need to be treated urgently) to construct the process ontology.

Following diagnostic and treatment processes of cancer at different stages

of malignancy pertaining to the cervix, vulva, uterus and ovary. A process map (Fig: B.1) can be found in the Section B.1 of Appendix B.

Operational Processes

- *Diagnosis process*: diagnosis code M13 combinations involving cervical cancer of the uteri and related codes.
- *Treatment process*: the 'treatment codes' manifested in the log without the description for the codes and corresponding treatment administered on the patients. Each patient may treated up to 16 treatment code. There are a total of 46 distinct treatment codes and 236 distinct treatment code combinations in the event log. A vast majority of treatment code combinations are unique combinations.

Compliance Processes

- *Activity process*: group based activities exhibit certain regularity compliances. The regularity is often manifested as a related set of diagnosis tests in the form of a continuous series of activities, e.g., different diagnosis blood tests prescribed for a patient in the lab.
- *Urgency Classifying process*: certain events in process are classified as urgent and non-urgent. Urgent cases are those cases where at least one activity of type urgent is manifested. There are a total 28 urgent activities. Processes with diagnosis code M11 (vulvar cancer) combination classified into urgent and non-urgent activities.

Audit Rule and its AROP Operational Audit Rules: "**Verify** the stages of (malignancy) cervical cancer in diagnosis code (M13)".

Following is the XML code based on *Reaction RuleML 0.2* that generated through Prova[a] rules engine.

```
1  <!-- file name: Healthcare_Verify.rrml -->
2  <?xml version="1.0" encoding="UTF-8"?>
3
4  <!-- This is a global active ECA reaction rules to
       verify the stages of a cervical cancer for a
       diagnosis code (M13) -->
5
6  <RuleML
7  xmlns="http://www.ruleml.org/0.91/xsd"
8  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
9  xsi:schemaLocation="http://www.ruleml.org/0.91/xsd
10 http://ruleml.org/reaction/0.2/rr.xsd">
11
12 <!-- Every stages actively detect "malignancyStage" if "
       squamous_cell_carcinoma (stages Ia1, Ia2, Ib, IIa,
       IIb, IIIb, IVa and IVb)" do "verifyCancerCervix"
13
14 ContractLog / Prova formalization (related to ISO Prolog
       notation)
```

[a]Prova ia an open source rule language for reactive agents and event processing https://prova.ws/index.html

```
15
16 eca(
17         everyStages(), % processing schedule for rule
18         detect(malignancyStage:event_stageType,stages),
              % event
19         squamous_cell_carcinoma (stages), % condition
              state
20         arop.healthcare.utils.verificationSystem.
              verifyCancerCervix(malignancyStage:
              event_stageType), %action
21         _, % empty post condition
22         _ % empty alternative action
23 ).
24 -->
25
26 <Assert>
27 <Rule style="active" evaluation="strong">
28
29  <!-- event -->
30
31  <on>
32   <!-- reaction rule with "clock" to detect the stages
       of cervical cancer. The rule it triggered by the
       everyStages event function and tries to detect the
```

```
     the malignancyStage event which becomes the trigger
      for the outer event of the ECA rule -->
33  <Rule style="active">
34   <on>
35        <Atom><Rel per="value">everyStages</Rel></Atom>
36   </on>
37   <do> <!-- raise new event -->
38        <Atom>
39         <Rel per="value">detect</Rel>
40         <Var type="event:stageType">malignancyStage</
               Var>
41         <Var>stages</Var>
42        </Atom>
43   </do>
44  </Rule>
45 </on>
46
47 <!-- condition -->
48
49 <if>
50  <Atom>
51   <Rel per="plain">squamous_cell_carcinoma</Rel>
52   <Var>stages</Var>
53  </Atom>
```

```
54  </if>
55
56  <!-- action -->
57
58  <do>
59   <Atom>
60    <!-- class/object -->
61    <oid><Ind uri="java://arop.healthcare.utils.
          verificationSystem"/></oid>
62    <!-- Boolean-valued static method -->
63    <Rel per="effect">verifyCancerCervix</Rel>
64    <!-- input parameter/argument -->
65    <Var type="arop.healthcare:stageType" mode="+">
          malignancyStage</Var>
66   </Atom>
67  </do>
68
69 </Rule>
70
71 </Assert>
72
73 </RuleML>
```

Compliance Audit Rules: "**Justify** the urgency of an activity: g0 (hemoglobin photoelectric)" that diagnosed with M13 and whose treatment code is 803.

Following is the XML code based on *Deliberation RuleML 1.01* that generated through Prova[b] rules engine.

```
1  <!-- file name: Healthcare_Justify.ruleml -->
2  <?xml version="1.0" encoding="UTF-8"?>
3  <?xml-model href="http://deliberation.ruleml.org/1.01/
       relaxng/datalog_relaxed.rnc"?>
4  <!--<?xml-model href="http://deliberation.ruleml.org
       /1.01/xsd/datalog.xsd" type="application/xml"
       schematypens="http://www.w3.org/2001/XMLSchema"?>-->
5
6  <RuleML
7  xmlns="http://ruleml.org/spec"
8  xmlns:xs="http://www.w3.org/2001/XMLSchema"
9  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
10 xsi:schemaLocation="http://ruleml.org/spec http://
       deliberation.ruleml.org/1.01/xsd/datalog.xsd">
11
12 <!-- A compliance audit rule using Unary predicates in
       datalog -->
13
14  <Assert mapClosure="universal">
15
16   <!-- i3 (Glucose), d1 (methemoglobin - sulphemoglobin
```

[b]Prova ia an open source rule language for reactive agents and event processing https://prova.ws/index.html

```
    each), b8 (bicarbonate), a5 (Calcium), b2 (Co-hb
    kwn), h4 (sodium - flame photometry), e7 (Potassium
     flame photometry), g0 (hemoglobin photoelectric)
    and b6 (Current ph - PCO2 - stand.bicarbonaat) are
    some of URGENT activities -->
17
18  <Atom>
19   <Rel>URGENT</Rel>
20   <Ind>e7 (Potassium flame photometry)</Ind>
21  </Atom>
22
23  <!-- Activity is URGENT if it is in diagnosis code (
    M13) and it's treatment code (803) -->
24  <Implies>
25   <then>
26        <Atom>
27         <op><Rel>URGENT</Rel></op>
28         <Var>x</Var>
29        </Atom>
30   </then>
31   <if>
32        <And>
33         <Atom>
34          <op><Rel>M13</Rel></op>
```

```
35              <Var>x</Var>
36             </Atom>
37             <Atom>
38              <op><Rel>803</Rel></op>
39              <Var>x</Var>
40             </Atom>
41            </And>
42     </if>
43    </Implies>
44
45   </Assert>
46
47 <!-- Can retract a fact which doesnot exist. The fact
       that g0 (hemoglobin photoelectric) is URGENT which
       does not exist is retracted. -->
48
49   <Retract>
50    <Atom>
51     <Rel>URGENT</Rel>
52     <Ind>g0 (hemoglobin photoelectric)</Ind>
53    </Atom>
54   </Retract>
55
56 </RuleML>
```

E-commerce System Processes

Event log of a loan application process [76]. The process represented in the event log is an application process for a personal loan or overdraft within a global financing organization. The event log contains events from three intertwined subprocesses, which can be distinguished by the first letter of each event name (**A**, **O** and **W**). The **A** subprocess is concerned with handling the applications themselves. The **O** subprocess handles offers send to customers for certain applications. The **W** process describes how work items, belonging to the application, are processed.

A and **O** are considered as operational and **W** considered as compliance subprocesses. In our experiment, we consider all three subprocesses as individual processes. A process map (Fig: B.3) can be found in the Section B.2 of Appendix B

Audit Rule and its AROP: Operational Audit Rules: "**Verify** the offers that has cancelled ("O_CANCELLED") after it was being sent "O_SENT" to the applicant"

Following is the XML code based on *Reaction RuleML 0.2* that generated through Prova[c] rules engine.

```
1 <!-- file name: Ecommerce_Verify.rrml -->
2 <?xml version="1.0" encoding="UTF-8"?>
3
```

[c]Prova ia an open source rule language for reactive agents and event processing https://prova.ws/index.html

```
4 <?xml-model href="http://reaction.ruleml.org/1.0/xsd/eca
    .xsd"?>
5 <RuleML xmlns="http://ruleml.org/spec" xmlns:xs="http://
    www.w3.org/2001/XMLSchema"
6 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7 xsi:schemaLocation="http://ruleml.org/spec http://
    reaction.ruleml.org/1.0/xsd/eca.xsd">
8 <!-- This defines a simple Event-Condition-Action rule
    which is triggered by an event pattern in the on part
    . A matching event instance is asserted which
    triggers the pattern in the ECA rule. The ECA rule
    proofs the condition and executes the action which
    asserts a new fact to the knowledge base. This fact
    is then queried. -->
9
10 <!-- assert rule -->
11
12  <Assert>
13
14   <formula>
15    <!-- rule is that - on the completion ``
        W_Nabellen_offertes\\COMPLETE" of pre-accepted ``
        A_PREACCEPTED\\COMPLETE" event of an application
        ``Case_ID" for assessment "W_Valideren_aanvraag\\
```

```
      START" if the assessment transmitted the
      application back to follow-up ``
      W_Nabellen_offertes\\START" do assert that
      System_Process cancels ``O_CANCELLED\\COMPLETE"
      the offers -->
16    <Rule style="active">
17     <on>
18      <Event>
19       <signature> <!-- define the event pattern for the
           detection -->
20        <Event>
21         <arg>
22                  <Expr>
23                   <op><Fun>W_Nabellen_offertes\\COMPLETE
                        </Fun></op>
24                   <arg><Var>Case_ID</Var></arg>
25                   <arg><Var>W_Valideren_aanvraag\\START
                        </Var></arg>
26                  </Expr>
27         </arg>
28        </Event>
29       </signature>
30      </Event>
31     </on>
```

```
32          <if>
33           <Equal>
34            <And>
35             <Atom>
36              <left><Var>W_Valideren_aanvraag</Var></left>
37              <right><Ind>START</Ind></right>
38             </Atom>
39             <Atom>
40              <left><Var>W_Nabellen_offertes</Var></left>
41              <right><Ind>START</Ind></right>
42             </Atom>
43            </And>
44           </Equal>
45          </if>
46          <do>
47           <Assert>
48            <formula>
49             <Atom>
50                    <op><Rel>O_CANCELLED\\COMPLETE</Rel></op
                        >
51                    <arg><Ind>System_Process</Ind></arg>
52                    <arg><Var>Case_ID</Var></arg>
53             </Atom>
54            </formula>
```

```
            </Assert>
          </do>
    </Rule>
  </formula>
</Assert>

<Assert>
  <!-- event "Case_ID: 174571 completes
       W_Nabellen_offertes\\COMPLETE then
       W_Valideren_aanvraag\\START and W_Nabellen_offertes
       \\START" -->
  <formula>
    <Event>
          <arg>
            <Expr>
        <op><Fun>W_Nabellen_offertes\\COMPLETE</Fun></op>
              <arg><Ind>Case_ID: 174571</Ind></arg>
              <And>
                <arg><Ind>W_Valideren_aanvraag\\START</Ind></
                    arg>
                <arg><Ind>W_Nabellen_offertes\\START</Ind></
                    arg>
              </And>
            </Expr>
```

```
74            </arg>
75        </Event>
76      </formula>
77
78    </Assert>
79
80  <!-- query knowledge base -->
81
82    <Query>
83      <Atom>
84        <Rel>O_CANCELLED\\COMPLETE</Rel>
85        <Ind>System_Process</Ind>
86        <Var>Case_ID: 174571</Var>
87      </Atom>
88    </Query>
89
90  </RuleML>
```

Compliance Audit Rules: "all cases must **follow** post-approval procedures (accept, register, and activate in sequence)"

Following is the XML code based on *Deliberation RuleML 1.01* that generated through Prova[d] rules engine.

```
1  <?xml version="1.0" encoding="UTF-8"?>
```

[d]Prova ia an open source rule language for reactive agents and event processing https://prova.ws/index.html

```
2 <?xml-model href="http://deliberation.ruleml.org/1.01/
    relaxng/folog_relaxed.rnc"?>
3 <!--<?xml-model href="http://deliberation.ruleml.org
    /1.01/xsd/folog.xsd" type="application/xml"
    schematypens="http://www.w3.org/2001/XMLSchema"?>-->
4 <RuleML xmlns="http://ruleml.org/spec"
5 xmlns:xs="http://www.w3.org/2001/XMLSchema"
6 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7 xsi:schemaLocation="http://ruleml.org/spec http://
    deliberation.ruleml.org/1.01/xsd/folog.xsd">
8
9 <!-- Integrity Constraints (IC) = {( Case_ID)approved(
    Case_ID) ( status) {accepted(Case_ID, True),
    registered(Case_ID, True), activated(Case_ID, Ture)}}
     -- A_APPROVED\\COMPLETE (approved) application (
    Case_ID), there exists O_ACCEPTED\\COMPLETE (accepted
    ) is True, A_REGISTERED\\COMPLETE (registered) is
    True, A_ACTIVATED\\COMPLETE (activated) is True -->
10
11 <Assert>
12
13 <Entails>
14
15  <Rulebase>
```

```
16 <!-- KB1 violets IC-->
17  <Atom>
18   <Rel>approved</Rel>
19   <Ind>Cased_ID: 176063</Ind>
20  </Atom>
21 </Rulebase>
22
23 <Rulebase>
24 <!-- IC -->
25  <Forall>
26   <Var>Case_ID</Var>
27   <Implies>
28    <Atom>
29     <Rel>approved</Rel>
30     <Var>Case_ID</Var>
31    </Atom>
32   <Exists>
33    <Var>status</Var>
34    <And>
35     <Atom>
36      <Rel>accepted</Rel>
37      <Var>Case_ID</Var>
38      <Var>status</Var>
39     </Atom>
```

```
40        <Atom>
41          <Rel>registered</Rel>
42          <Var>Case_ID</Var>
43          <Var>status</Var>
44        </Atom>
45        <Atom>
46          <Rel>activated</Rel>
47          <Var>Case_ID</Var>
48          <Var>status</Var>
49        </Atom>
50       </And>
51     </Exists>
52     </Implies>
53    </Forall>
54  </Rulebase>
55
56 </Entails>
57
58 <Entails>
59
60  <Rulebase>
61  <!-- KB2 obeys IC-->
62    <Atom>
63      <Rel>approved</Rel>
```

```
64      <Ind>Cased_ID: 176063</Ind>
65    </Atom>
66    <And>
67      <Atom>
68        <Rel>accepted</Rel>
69        <Ind>Cased_ID: 176063</Ind>
70        <Data>True</Data>
71      </Atom>
72      <Atom>
73      <Rel>registered</Rel>
74        <Ind>Cased_ID: 176063</Ind>
75        <Data>True</Data>
76      </Atom>
77      <Atom>
78        <Rel>activated</Rel>
79        <Ind>Cased_ID: 176063</Ind>
80      <Data>True</Data>
81      </Atom>
82    </And>
83  </Rulebase>
84
85  <Rulebase>
86  <!-- IC -->
87    <Forall>
```

```
88    <Var>Case_ID</Var>
89    <Implies>
90     <Atom>
91      <Rel>approved</Rel>
92      <Var>Case_ID</Var>
93     </Atom>
94     <Exists>
95      <Var>status</Var>
96     <And>
97      <Atom>
98       <Rel>accepted</Rel>
99       <Var>Case_ID</Var>
100      <Var>status</Var>
101     </Atom>
102     <Atom>
103      <Rel>registered</Rel>
104      <Var>Case_ID</Var>
105      <Var>status</Var>
106     </Atom>
107     <Atom>
108      <Rel>activated</Rel>
109      <Var>Case_ID</Var>
110      <Var>status</Var>
111     </Atom>
```

```
112         </And>
113         </Exists>
114       </Implies>
115     </Forall>
116   </Rulebase>
117
118 </Entails>
119
120 </Assert>
121
122 </RuleML>
```

Production Monitoring System Processes

An incident and problem handling system of Volvo IT which is called VINST [70] that mainly supports the two following processes:

- *Handle Incidents Process*: the primary goal of this process is to restore normal service operation (Normal service operation' is defined within Service Level Agreement (SLA)) as quickly as possible and by that ensuring the best possible levels of service quality and availability are maintained. Incidents that cannot be resolved by the Service Desk or Expert Helpdesk should be escalated to Second Line and/or Third Line teams. Solution should be established as quickly as possible in order to restore the service to normal with minimum disruption to the business.

After implementing a Solution by IT departments (and specialist teams) and *verifying that the service is restored* the Incident is closed. If the Action Owner suspects that the Incident might reoccur a *Problem record shall be registered*.

- *Handle Activity Problem Process*: it describes how to handle Problems in the IT-services delivered and/or operated by Volvo IT. This process diagnoses the root cause(s) incidents activities and secures the resolution of those problems to enhance the quality of IT-services delivered and/or operated by Volvo IT.

  Handle Activity Problem Process works together with other processes like Handle Incidents Monitor service, Discover & Define Opportunity, Develop, Deploy & Provide and Manage Service Change etc. to ensure that IT service availability and quality are increased.

  Handle Activity Problem Process should also, when applicable, *verify and update Solutions in the knowledgebase*, so that the best possible Solution is available during the life-cycle of the problem. Two process maps (Fig: B.5, B.6) can be found in the Section B.3 of Appendix B.

Audit Rule and its AROP Control Monitoring Audit Rules: "Monitor the users that accept the Wait-User substatus".

Following is the XML code based on *Reaction RuleML 0.2* that generated through Prova[e] rules engine.

```
1 <!--  file name: Production_Monitor.rrml -->
```

[e]Prova ia an open source rule language for reactive agents and event processing https://prova.ws/index.html

```
2  <?xml version="1.0" encoding="UTF-8"?>
3  <?xml-model href="http://reaction.ruleml.org/1.02/
       relaxng/kr-cep.rnc" type="application/relax-ng-
       compact-syntax"?>
4  <RuleML xmlns="http://ruleml.org/spec" xmlns:xs="http://
       www.w3.org/2001/XMLSchema"
5  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6  xsi:schemaLocation="http://ruleml.org/spec http://
       reaction.ruleml.org/1.02/xsd/cep.xsd">
7  <!--
8  This defines a simple Event-Condition-Action rule which
       is triggered by an event pattern in the on part. A
       matching event instance is asserted which triggers
       the pattern in the ECA rule. The ECA rule proofs the
       condition and executes the action which asserts a new
        fact to the knowledge base. This fact is then
       queried.
9  -->
10
11   <!-- assert rule -->
12
13   <Assert>
14   <!-- rule is that - "on receiving the assignment of a
        Resource for a Concept if the Concept is 'Accepted'
```

```
      do tell Resource (=send to Resource message) that
      Incident_Manager monitors the Resource" -->
15      <Rule style="active">
16        <on>
17          <Receive> <!-- receive action waiting for
                incoming messages -->
18            <enclosed>
19              <Message>   <!-- message pattern definition
                    matching against incoming messages -->
20                <signature> <!-- define the message's
                    signature as event pattern for the
                    detection -->
21                  <Atom>
22                    <Rel>assigns</Rel>
23                    <Var>Resource</Var>
24                    <Var>Concept</Var>
25                  </Atom>
26                </signature>
27              </Message>
28            </enclosed>
29          </Receive>
30        </on>
31
32        <if>
```

```
33              <Atom>
34               <Rel>Accepted</Rel>
35               <Var>Concept</Var>
36              </Atom>
37             </if>
38
39             <do>
40              <Send>
41               <enclosed>
42                <Message>
43                 <receiver><Var>Resource</Var></receiver>
                      <!-- receiver is THE "Person" -->
44                 <payload> <!-- payload of message is a
                      RuleML knowledge base -->
45                  <RuleML>
46                   <Assert>
47                    <Atom>
48                     <Rel>monitors</Rel>
49                     <Ind>John:Incident_Manager</Ind>
50                     <Var>Resource</Var>
51                    </Atom>
52                   </Assert>
53                  </RuleML>
54                 </payload>
```

```
55                </Message>
56               </enclosed>
57              </Send>
58             </do>
59           </Rule>
60
61           <Atom>
62            <Rel>Accepted</Rel>
63            <Ind>Wait-user</Ind>
64           </Atom>
65
66   </Assert>
67
68   <!-- send message that "Resource_ID assigns Wait-user
        status" which triggers the messaging reaction rule
        which is waiting for assignments -->
69
70   <Send>
71    <enclosed>
72     <Message>
73      <cid><Ind>cid1</Ind></cid> <!-- conversation
           identifier -->
74      <sender><Ind>Resource_ID</Ind></sender> <!-- sender
           -->
```

```
75        <receiver><Ind>John:Incident_Manager</Ind></
              receiver> <!-- receiver -->
76        <payload> <!-- payload -->
77         <RuleML>
78          <Assert>
79           <Atom>
80            <Rel>assigns</Rel>
81            <Ind>Resource_ID</Ind>
82            <Ind>Wait-User</Ind>
83           </Atom>
84          </Assert>
85         </RuleML>
86        </payload>
87       </Message>
88     </enclosed>
89    </Send>
90
91 </RuleML>
```

9.4.3 Results and Interpretations

Rules using The Rule Markup Language (RuleML) describe the general association of causes with effects ('laws'), situations with actions ('triggers'), premises with conclusions ('implications'), and so are used to represent physical, chemical and biological processes, medical guidelines, business and legal

policies, conditional equations, probabilities and preferences, grammars, logics, database views, and declarative programs. Reaction RuleML and Deliberation RuleML are two of the many branches of RuleML. Reaction RuleML for action and/or reactions. Reaction rules subsume Complex Event Processing (CEP) and Knowledge Representation (KR) rules, as well as Event-Condition-Action-Postcondition (ECAP) rules. ECAP rules specialize to Event-Condition-Action (ECA) rules, which themselves specialize to Condition-less Trigger (EA) rules and to the rule subfamily of Event-less Production (CA) rules.

Deliberation RuleML for inference. Deliberation rules, via Higher Order Logic (HOL) and First Order Logic (FOL), subsume Derivation rules. Derivation rules subsume Hornlog and Datalog languages and (syntactically) specialize to the condition-less Fact and conclusionless Query languages (subsuming Integrity Constraint (IC) languages). Recently, Deliberation RuleML 1.01 has developed the ability to combine one or more of the Existential Rules, Equality Rules, Integrity Rules Datalog extensions which together define $Datalog^{+}$.

Operational audit rules were developed using Reaction RuleML and compliance audit rules were using Deliberation RuleML. Experimental results of the above audit rules are presented in two categories: operational and compliance auditing.

Operational Auditing

Healthcare System: Healthcare dataset contains 150,291 events in over 1143 processes. Audit rule - "**Verify** the stages of (malignancy) cervical cancer in diagnosis code (M13)". Cancer "malignancy" levels were detected in every stages of "squamous_cell_carcinoma" in diagnosis code (M13) that verifies the

cervical cancer. There are 8 stages of "squamous_cell_carcinoma" (Ia1, Ia2, Ib, IIa, IIb, IIIb, IVa and IVb) to determine the "malignancy" level. Audit rule have triggered action on a total of 252 processes in the event log satisfying the condition of "squamous_cell_carcinoma" stages criteria. We have also found that in this 252 processes, there are 14611 events distributed over 272 activities. Only 03 (three) processes were detected where activity $a0$ (CEA - tumor marker using meia) were found before $e7$ (squamous cell carcinoma using eia).

E-commerce System: E-commerce event log is comprised of a total of 262,200 events within these 13,087 processes, starting with a customer submitting an application and ending with eventual conclusion of that application into an Approval, Cancellation or Rejection (Declined). Audit rule - "**Verify** the offers that has cancelled ("`O_CANCELLED`") after it was being sent "`O_SENT`" to the applicant". This is a knowledge based audit rule which means this rule learns the behavior from predefined query knowledge base then it asserts that knowledge condition to the events. Out of 13,087 processes, this audit rule found out 18 processes (or loan applications) were cancelled after the loan offer was sent to the applicant. We can make an assumption which could be system fault or uncareful assessment that led them to the cancellation. Average application span was 8days 15hrs and 21min. Shortest span was 17hrs 24min and longest was 15days 15hrs.

Compliance Auditing

Healthcare System: There are activities that are classified as urgent and ordinary counterparts to such activities also exist. Audit rule - "**Justify** the

urgency of an activity: g0 (hemoglobin photoelectric)" that diagnosed with M13 and whose treatment code is 803. There are a total of 28 urgent activities in the event log. Table 9.4 depicts the evidence pertaining to the processes in the event logs that diagnosed with M13 and whose treatment code is 803.

Process events	0481	0257	0560	0499	0466	0058	0683	0619
g8	g8	g8	–	g8	g8	g8	–	–
–	–	b8	–	d5	–	–	–	–
–	–	f0	–	f0	–	–	–	–
–	–	g0	–	c2	–	–	–	–
e1	e1	–	e1	e1	e1	e1	e1	e1
a8	a8	–	a8	a8	a8	a8	a8	a8
e0	e0	–	e0	e0	e0	e0	e0	e0
f8	f8	–	f8	f8	f8	f8	f8	f8
j3	j3	–	j3	j3	j3	j3	j3	j3
c9	c9	c9	c9	c9	c9	c9	c9	c9
d6	d6	d6	d6	d6	d6	d6	d6	d6
g7	g7	–	g7	g7	g7	g7	g7	g7
b5	b5	–	b5	b5	b5	b5	b5	b5
c4	c4	–	c4	c4	c4	c4	c4	c4
h0	h0	h0	h0	h0	h0	h0	h0	h0
c0	c0	–	c0	c0	c0	c0	c0	c0
g5	g5	–	g5	g5	g5	g5	g5	g5
a0	a0	–	a0	a0	a0	–	a0	a0
–	e7	–	–	e7	–	g0	a5	g0
f3	f3	–	f3	f3	f3	f3	f3	f3
h2	h2	–	h2	h2	h2	h2	h2	h2
e4	e4	e4	e4	e4	e4	e4	e4	e4
a6	a6	–	a6	a6	a6	a6	a6	a6
f2	f2	f2	f2	f2	f2	f2	f2	f2
d5	–	–	d5	f2	f2	d5	d5	d5
f0	f0	f0	f0	–	f0	f0	f0	f0
c2	c2	–	c2	–	c2	c2	c2	c2

Table 9.4: Urgent processes diagnosed with M13 and whose treatment code is 803.

Some of the observations are:

The event 'g0 (hemoglobin photoelectric)' classified the processes 0058, 0257, and 0619 as urgent and they are retracted as urgent processes by the audit rule. Presence of 'e7 (Potassium flame photometry)' event also classified the processes 0481, and 0499 as urgent processes.

Processes 0560, 0466, and 0683 are also classified as urgent processes but they neither retracted nor classified by the audit rule. Hence, the reason/evidence needs to be found out why those processes were classified as urgent processes.

Process 0257 skips a lot activities because of either urgency level was severe or further diagnosis was deem unnecessary.

Somewhat all the urgent processes start with g8 (nursing gynecology short-out) and instant opinion of nurse is very important in this g8 (nursing gynecology short-out) event.

E-commerce System: Audit rule - "all cases must **follow** post-approval procedures (accept, register, and activate in sequence)". There are three event (`O_ACCEPTED, A_REGISTERED, A_ACTIVATED`) must be followed after application approval process. There was no specific guidelines to complete the events either sequentially or randomly. Observations are: after using the audit rule is that all three post-approval (`O_ACCEPTED, A_REGISTERED, A_ACTIVATED`) events occur in random order for 2,246 `A_APPROVED` applications. 3 applicants completed `A_REGISTERED` before completing `O_ACCEPTED`. Audit rule found these 3 applicants as `A_ACTIVATED\\COMPLETE` because `A_REGISTERED\\COMPLETE` learned the `O_ACCEPTED\\COMPLETE` status as well.

Rule Based Control Monitoring

Production System: Knowing that there are a lot of KPIs measuring the total resolution time of an incident people try to find workarrounds that stop the clock from ticking. One way of doing this is manually giving an incident the substatus Wait User. Although there are guidelines not to use this substatus (unless someone is really waiting for an end-user), some people (action owners) are breaking this guideline (see Table 9.5 below).

Resource	Resource Country	Frequency	Relative Frequency
Andreas	Sweden	40	9.22%
Cezary	Poland	25	5.76%
Emil	Sweden	42	9.68%
Jinos	India	23	5.30%
Olga	Poland	47	10.83%
Muthu	India	76	17.51%
Natalia	Brazil	27	6.22%
Nina	Poland	27	6.22%
Oden	Sweden	21	4.84%
Pawel	Poland	58	13.36%
Rafal	Poland	28	6.45%
Vandana	India	20	4.61%

Table 9.5: Users that accepted the Wait-User substatus.

Audit rule - "Monitor the users that accept the Wait-User substatus" to track all the users that using this substatus. This Complex Event Processing (CEP) rule which is triggered by 'Accepted' status event pattern. A matching event 'Wait-user' instance is asserted which triggers the pattern in the ECA rule. The ECA rule proofs the condition and executes the action which asserts a new fact 'Resource_ID' to the knowledge base. This fact is then queried. This audit rule monitors the fact 'Resource_ID' that changes the status of a

'Concept' to 'Wait-user' substatus instantly.

9.5 Discussion of Results and Concluding Remarks

To solve the problem of accessing heterogeneous distributed data sources, our proposed Common Ontology and Process Ontology within distributed environment along with audit rules and AROP has been experimented with using three datasets from three different heterogeneous pervasive systems. A total of 20602 instances of processes was used to experiment with the accessibility of heterogeneous data sources. In this chapter, we have also evaluated the audit rules and CPA in operational auditing, compliance auditing, and control monitoring.

The 'Verify' audit rule was enforced with respect to the healthcare and e-commerce datasets for operational auditing to detect certain attributes of event(s) in a process. Operational audit rule and sub-rules were implemented using Reaction RuleML specification's Event-Condition-Action (ECA) rules and Knowledge Representation (KR) rules. These ECA and KR based audit rules enforce action as the condition satisfy in the events. In the same way, 'Justify' audit rule was applied in both healthcare and e-commerce datasets for compliance auditing to check the compliances of certain requirements in a process. Compliance audit rule and sub-rules were implemented using Deliberation RuleML specification's Derivation rules and Integrity Constraints (IC) query language.

'Monitor' audit rule was enforced in production dataset for Continuous Control Monitoring that sends message to the stockholders by inferring knowl-

edge in the event attributes and asserting action in the form of message. Reaction RuleML's Complex Event Processing (CEP) rules specification were used to implement the CCM audit rules.

Our proposed hybrid layered ontological approach to solve the problem of accessing heterogeneous distributed data sources has provided the accessibility with few exceptions. Three specific audit rules and their AROP with POs and respective Common ontologies has developed to test our hypotheses in three different datasets of pervasive systems. 'Verify' and 'Justify' audit rules were used for both operation and compliance process auditing and 'Monitor' audit rule was used to monitor the certain behaviours of production system and to send messages to the stakeholders. These three audit rules in conjunction with their AROPs, POs and Common Ontologies have detected the behaviours in the events and have enforced actions that outlined in the audit rules.

We have experimented our novel hypotheses and methodologies in three different heterogeneous pervasive systems that showed the promising results and the applications of our benchmark Continuous Process Auditing (CPA) system for operational and compliance auditing. Due to the unavailability of similar kind of CPA system within both research and for-profit communities, we were unable to evaluate and measure the success ratio and advancement of our benchmark CPA system with other similar systems. We would love to participate and share our source codes to evaluate with other such systems and extend future research opportunities

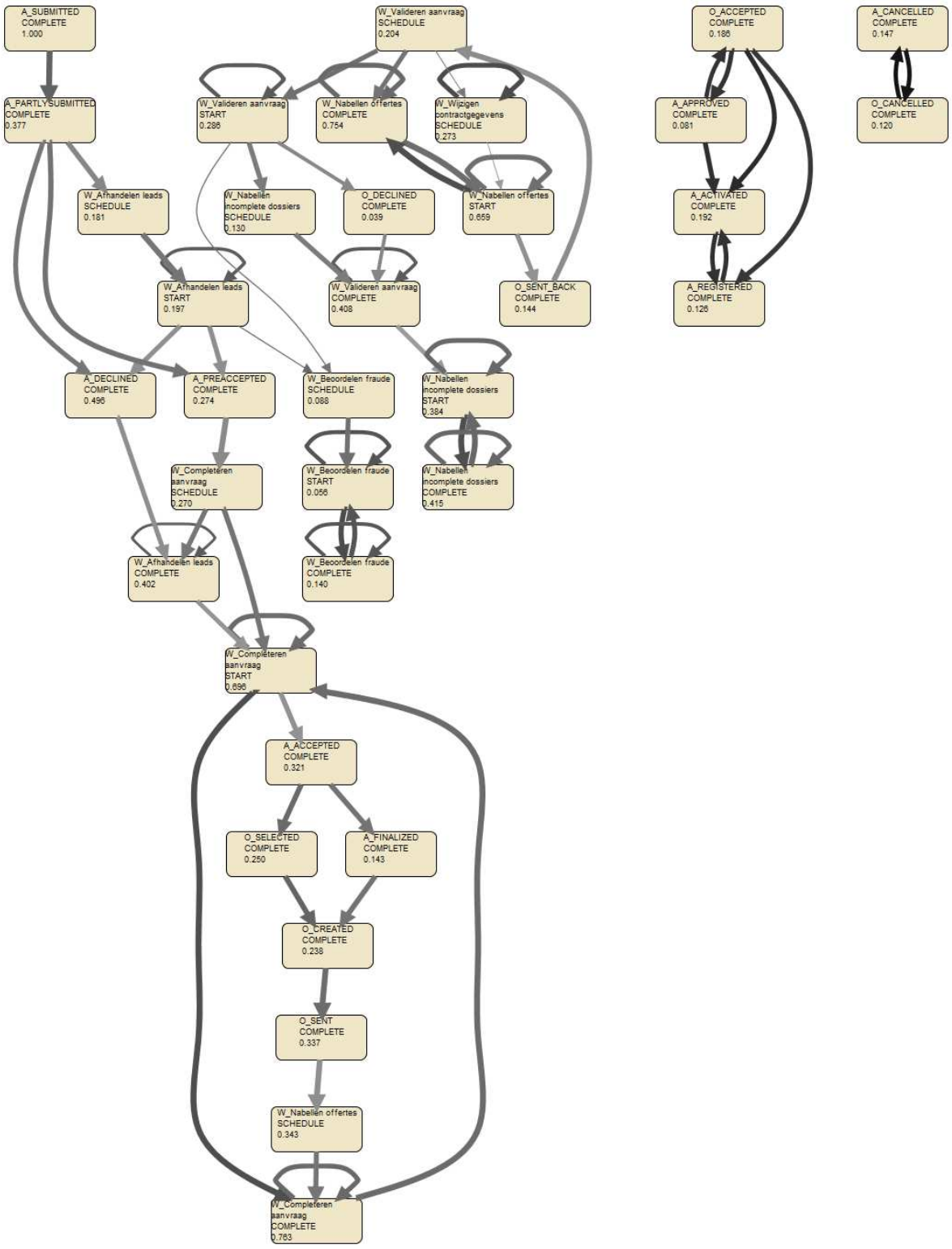


Figure 9.2: Dutch Financial Institute - personal loan application *Process Maps*

Part IV

Conclusion

Chapter 10

Reformulation of the Continuous Process Auditing (CPA) Problem

Though our main focus has been the development of methodologies for solving the Continuous Process Auditing (CPA) problem, and subsequent issues related to CPA, in this chapter we discuss some thoughts and directions that have arisen directly from our research. In as much as these thoughts require substantially more work beyond the scope of this dissertation, yet have become clearly identified as a direct result of our current work. We have not incorporated them into this dissertation; rather, we present them as problems that are open for future research opportunities.

From 'fact' based audit rule to automatization of audit rule, knowledge acquisition, AROP generation to process mining in CPA to evidence evaluation and predicting recommendations, several avenues of thinking and analysis have become more clear. In the following sections, we have tried to list most of

these thoughts and directions which merit immediate attention, in our opinion.

10.1 Defeasible Logic and Facts

Description Logics (DL)[a] is a family of formal knowledge representation languages. Many DLs are more expressive than propositional logic, but less expressive than first-order predicate logic. In contrast to the latter, the core reasoning problems for DLs are (usually) decidable, and efficient decision procedures have been designed and implemented for these problems. A Descriptive Logic (DL) models *concepts, roles and individuals*, and their relationships. The fundamental modeling concept of a DL is the *axiom* - a logical statement relating roles and/or concepts. This is a key difference from the frames paradigm where a *frame specification* declares and completely defines a class.

We have constructed audit rules that describe the general association of causes with effects ('laws'), situations with actions ('triggers'), premises with conclusions ('implications'). Operational audit rule and sub-rules implemented using Reaction RuleML specification's Event-Condition-Action (ECA) rules and Knowledge Representation (KR) rules. These ECA and KR based audit rules enforce action as the condition satisfy in the events. Compliance audit rule and sub-rules implemented using Deliberation RuleML specification's Derivation rules and Integrity Constraints (IC) query language.

Defeasible logic is a simple and efficient rule-based non-monotonic formal-

[a] https://en.wikipedia.org/wiki/Description_logic

ism that derives plausible conclusions from partial, and sometimes conflicting, information. The knowledge in a *Defeasible Theory*[b] is organised in *facts* and *rules* and *superiority relation*. Rules are divided into *strict rules, defeasible rules* and *defeaters*.

Audit rules constructed for operational and compliance auditing are expressive in nature. These DL based audit rules can form implications in association with causal effects and situational actions. However, they cannot form or find any facts. What are the facts? Facts are indisputable statements. Finding facts in rule-based Continuous Process Auditing is yet to be investigated by the research community.

10.2 Towards Audit Rule to AROP Generation and Stemming of AROPs

Stemming, also known as branching, is the process for reducing inflected or derived words to their stem or root, generally a written word form. Stemming is also a very powerful technique for mapping related words to the same stem, or root.

So far as part of this dissertation research, we have designed and implemented the Healthcare Common Ontology as the top-most layer in a Hybrid Layered Ontology towards the generation of Audit Rule Ontology of a Process as a second layer. We have identified the need to devise a specific semantic similarity technique and we have been mechanizing a natural language processing (NLP) technique, based on stemming algorithms, to add the AROPs

[b] http://www.defeasible.org/PhD/2014/FrancescoOlivieri.pdf - Sec 2.1

to the Healthcare Common Ontology. In particular, we are considering the hybrid approach or stochastic probability type of stemming algorithm to identify the root or stem from the top-most layer Healthcare Common Ontology to generate the AROPs, and to add the AROPs as a second layer.

10.3 Knowledge Acquisition and Engineering

Acquiring rules is always a interesting problem in Natural Language Processing (NLP). Our proposed audit rule approach is lenient to Descriptive Logic's family of formal knowledge representation languages. We need to devise an acquisition methodology that first acquires the axioms from defined audit rules, then transforms the axioms to RIF or RDF format. It would be interesting to investigate the applications of this approach in the following, yet to be explored, areas:

1. Rule based knowledge acquisition and engineering
2. Process based data fusion and process discovery
3. Predictive and prescriptive evidence generation

10.4 Rule Automation

Our CPA approach allows auditors to write audit rules in Natural Language (i.e English) either in abstract or elaborative manner. One of the challenges we faced was converting audit rules from natural language to Descriptive Logics (DL) to the form of RuleML. So far we have done the process manually.

It would be interesting to explore the ideas of automatizing the audit rule translation from English (natural language), to RuleML format. This rule automation process would augment how human audit professionals write audit rules in English (more natural languages may be considered later) for direct use in Continuous Process Auditing.

10.5 Evidence Evaluation and Predicting Recommendations

A major challenge in auditing is determining whether the audit evidence obtained is sufficient and appropriate to support the opinion to be expressed. Making sense of the evidence gathered is appropriate (relevant, reliable and valid), which is persuasive rather than convincing and sufficient to support assertions and to provide assurance. What evidence need to collect and/or retrieve, from where (what are the sources), and when (determination of the relevance and reliability)? Developing an evidence evaluation technique that determines the sufficiency and appropriateness of evidence. That means evaluation technique follows through to its logical conclusion throughout the whole process without being biased.

So far we have developed the methodologists to detect evidence. Whether the detected evidences are sufficiently appropriate is an open problem yet to investigate. At the same time, tool or methodologies for predicting recommendations in the form of ethical guidance, compliance, and operational rules is also need to be investigated. Developing recommendation tool that gen-

erates appropriate and sufficient recommendations to optimize the evidence collection and evaluation procedures in the total auditing process of Continuous Process Auditing is another streams of open research problems yet to be investigated.

10.6 Process Mining in Continuous Process Auditing

Our approach has shown some opportunities for the future of CPA. Event based processes were used to experiment with our proposed CPA methodologies. The challenges and opportunities of process discovery [83, 44], and process mining of event logs [40, 41] in auditing are already being addressed by researchers. Rule based process mining of event logs in Continuous Process Auditing is another fruitful area to be explored; this has particular relevance to semantic reasoning and DLs (see Section 10.1).

10.7 Continuous Process Auditing-as-a-Service

The emerging growth and evolution of web based systems and services make the job of audit professionals a complicated and time-consuming one for many enterprises. In this context, continuous process auditing (CPA) systems in the form of audit-as-a-service (AaaS) emerges as an inexpensive and effective approach. A CPA system helps to satisfy process auditing needs and recommendations in the context of distributed enterprise systems while requiring fewer resources and enabling processes to be audited continuously in real-time.

Subhani et. al. [72] proposed a conceptual architecture for Continuous Process Auditing (CPA) based on domain ontologies, audit rules, knowledge learning techniques and audit report recommendation procedures. They have sketched a representation (see Figure 10.1) of a CPA system for a process based e-commerce platform, offering customizable audit rule based solutions for audit professionals, system administrators and senior decision makers. This service based system is yet to be implemented and can be investigated in consideration with the following aspects.

Audit rules may be generated in several different situations with various constraints; including such as, ontological association rules, predefined sets of compliance rules, environmental factors, on-demand auditing matter, and so on, using switching to support a hybrid recommender system. Various data mining techniques may be incorporated with recommender systems to generate more accurate audit rules by applying these techniques to historical evidential data to predict and classify them in order to make improved clarified decisions. These recommender systems and data mining techniques can be adapted to fulfill client demands and requirements.

10.8 Big Data and Transforming to the Predictive Auditing Analytics

The implementation of CA is a recognized challenge among researchers and practitioners, and traditional audit tools and techniques neglect the potential of Big Data Analytics. Predictive analytics does not tell us what will

happen in the future; rather, it tries to predict possible future outcomes and trends. Using Big Data and Big Data computing, Continuous Auditing and Control Monitoring would drive the future of Predictive Auditing Analytics (PAA). Predictive Auditing Analytics model would be future thinking that drive with a periscope to detect going concerns and fraudulent activities which adds values to the business profitability. In 2014, AICPA envisioned creating opportunities for researchers and audit professionals to develop Predictive Auditing Analytics and Audit Data Analytics tools and services for the Big Data era.

10.9 Summary

Our proposed hypotheses and methodologies have developed and tested in the Chapters 6, 7, 8, and 9. In this chapter we discuss some thoughts and directions that have arisen directly from our research of solving the Continuous Process Auditing (CPA) problem, and subsequent issues related to CPA. These problems including, not limited to, from 'fact' based audit rule to automatization of audit rule, knowledge acquisition, AROP generation to process mining in CPA to evidence evaluation and predicting recommendations, several avenues of thinking and analysis have become more clear. Since these thoughts require substantially more work beyond the scope of this dissertation; rather, we present them as problems, in our opinion, that are open for future research opportunities.

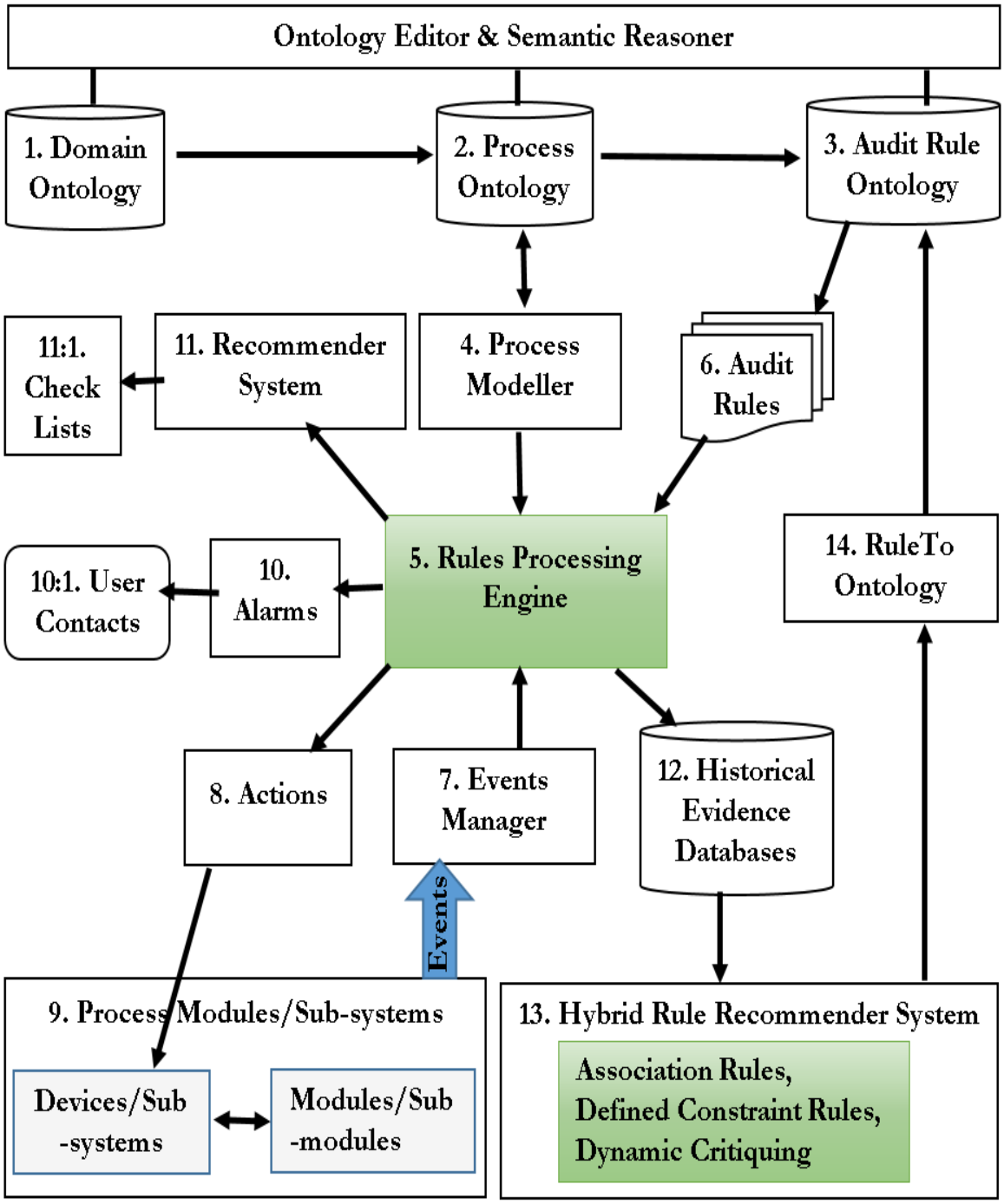


Figure 10.1: Architecture of a Continuous Process Audit-as-a-Service (AaaS).

Chapter 11

Summary and Conclusion

In this chapter, we summarize the work that has been done in this thesis. In the Summary section, we have sketched out the part-by-part description of all previous parts. The Conclusion lists our contributions.

11.1 Summary

11.1.1 Background and Foundations (Part I):

In this part of the thesis, we established the problem definition and foundations of Continuous Process Auditing research, along with its open issues, challenges and potential research methodologies. Background and literature studies of auditing are presented in Chapter 2. A taxonomic analysis of Continuous Auditing and its review is discussed in Chapter 3. Comparison of Continuous Auditing vs Continuous Monitoring and the necessity of CPA in Compliance and Operational auditing is presented in Chapter 4. In Chapter

5, we analyzed the different models and frameworks of Continuous Auditing System that already exist or are under development in industry, as well as in the research community. We have also presented a comparative analysis of CA methodologies from the perspectives of practicality and potentiality. Finally, we presented some Open Issues and challenges of Continuous Process Auditing and its methodologies.

11.1.2 Methods for Continuous Process Auditing (Part II):

Proposed CPA methodologies, including algorithms, implementation mechanisms and proposed framework, were listed. Domain of CPA and its processes are the two main components of our methodologies. Knowledge representation mechanisms of domain and its processes are presented in Chapter 6. We devised a two layered ontological approach for domain and process knowledge presentation. Common Ontology (CO) forms the top layer to serve all shareable common vocabulary and Process Ontology for each processes are constructed in second layer. This CO and POs' forms a common ontological layer to assure the seamless heterogeneous data accessibility between domain and its processes.

In Chapter 7, we devised an audit rule based ontological approach for CPA. A collection of audit rules for a process are required to construct an Audit Rule Ontology of a Process (AROP). Each AROP belongs to a specific process. AROPs are mapped to Common Ontology (CO) and POs to construct a hybrid layered ontology. Construction and mapping mechanisms and algorithms also presented in this chapter.

11.1.3 Evaluation and Applications (Part III):

Evolution of Audit Rule Ontology and evaluation of CPA methodologies are expressed in this Part III. In Chapter 8, we proposed a method that updates the previous version of an ontology. It is a process of evolution methodology within the domain of ontology lifecycle that creates newer versions by stemming new branches to the original ontology.

In Chapter 9, we evaluated the Continuous Process Auditing and its Audit rules, and Audit Rule Ontology of a Process (AROP). Processes were either mined or used from three different datasets of pervasive systems: healthcare, e-commerce and production management. For each of the systems, we constructed the Domain Ontology and Process Ontology for one specific process. Audit Rules were written in an abstract manner for Operational Auditing and Compliance Auditing; then, rules were converted to RuleML. Both operational and compliance audit rules provided for experiments in healthcare and e-commerce systems. Rule Based Control Monitoring were experimented with in Volvo IT production management system. Operational audit rules were converted according to the specifications of Reaction RuleML and Compliance audit rules were converted according to the Deliberation RuleML. A total of 20602 instances of three processes (one from each datasets) have used to experiment audit rules and their AROPs, Process Ontology and Domain Ontology by accessing events from the heterogeneous distributed data sources through database access protocols like ODBC, JDBC, OData, and MySQL.

Outcomes and results of experiments are also presented with review and analysis at the end of Chapter 9.

11.1.4 Reformulation of the CPA Problem and Conclusion (Part IV):

We presented two chapters in this part IV: the reformulation of the CPA problem, and conclusion of this dissertation. In Chapter 10, we have reformulated the whole CPA problem in the context of thoughts and directions that have arisen directly from our research. These research issues and ideas have not been incorporated into this dissertation and we leave them open for future research opportunities.

Our proposed AROP in chapter 7 and audit rules constructed for operational and compliance auditing are expressive in nature. These Descriptive Logic (DL) based audit rules can form implications in association with causal effects and situational actions. They cannot form or find any facts. What are the facts? Facts are indisputable statements. Finding facts in rule based Continuous Process Auditing can be investigated by designing the specifications for Defeasible Logic which is a tuple of *facts*, *rules* and *superiority relation*.

Audit rules are usually written in natural language (e.g. English), then AROPs are constructed. These AROPs are stemmed to the Common Ontology as a second layer. Forming audit rules to constructing AROP to stemming AROP into Common Ontology are done manually. As a first step towards automating this procedure, rule acquisition from human written audit rules is a vital necessity. Rule automation from natural language to the form of Rule Language (i.e. RuleML, XRML, SWRL) is a problem of natural language processing where identification of audit rule is required as well as rule axiomatization. Automatization of whole procedure should be addressed accordingly

towards the adaptation of our proposed approaches.

Detecting evidence that is sufficient and appropriate to support the opinion is necessary to predict evidence based recommendations. Making sense of detected evidence is relevant, reliable and valid that sufficiently supports the assertions and provides assurance. Under this evidence trajectory, evidence retrieval mechanism from various heterogeneous sources and developing evidence evaluation technique that is sufficeint and appropriate to assert/predict logical conclusion throughout the whole process without being biased any steps. Evidence logs and rule based process mining of event logs in Continuous Process Auditing is another way to look into the evidence detection.

In the era of emerging growth of EIS and web based systems, providing service via alternative media is needed. Web based auditng architecture and tool could be provided as a service [72]. Continuous process auditing (CPA) systems in the form of audit-as-a-service (AaaS) emerges as an inexpensive and effective approach. A CPA system helps to satisfy process auditing needs and recommendations in the context of distributed enterprise systems while requiring fewer resources and enabling processes to be audited continuously in real-time. Quick adaptability to various situations and constraints; including such as, ontological association rules, predefined sets of compliance rules, environmental factors, on-demand auditing matter, recommender systems, data mining techniques, and so on, make AaaS more reliable to fulfill client demands and requirements.

As an aspect of thinking towards adapting audit to the future technology, Predictive Auditing Analytics (PAA) and Big Data Computing would drive potential detection of concerns and fraudulent activities, which adds value to

the business. It would also help to create opportunities for researchers and audit professionals to develop Predictive Auditing Analytics and Audit Data Analytics tools and services, such as envisioned in 2014 AICPA report [19].

11.2 Conclusion

Continuous Auditing of a Process, or Continuous Process Auditing, in a pervasive system remains a research challenges yet to addressed in full. We have endeavoured to solve part of the CPA research challenge, namely: (i) heterogeneous distributed access; (ii) knowledge representation of domain and processes; and (iii) providing audit and control monitoring assurance. We have presented each of these research problems first individually, and then collectively, to solve the set of problems together in a continuous event based manner.

First, we have devised a hybrid model approach to construct a hybird layered ontology (Common Ontology). This Common Ontology is built on by combining two ontologies: Domain Ontology and Process Ontology. Domain ontology defines and contains the vocabulary of the structure and accessibility information of a specific domain. Process ontology defines and contains the vocabulary of the structure and sequential access information of a process in a domain. Domain ontology is the top layer and process ontology is the second layer in the hybrid layered Common Ontology (CO). This CO facilitates the seamless accessibility to the heterogeneous distributed data sources and also facilitates to store the knowledge vocabulary of domain and processes. We have also presented the evolution mechanisms to update the ontologies as

they are required to evolve with the changing business needs. Audit rules were DL expressive and have been implemented to trigger and to enforce action on detected events.

Secondly, we have presented the rule based audit approach for Continuous Process Auditing. An audit rule sheet in traditional auditing defines all the audit rules for a specific process. We have converted the audit rule sheet to an Audit Rule Ontology of a Process (AROP). All audit rules from an audit rule sheet are put together to construct an AROP, then the AROPs were mapped to PO then to CO. Each rule is constructed using Description Logic (DL) specifications that are described in Reaction RuleML and Deliberation RuleML. Event-Condition-Action and Integrity Constraints based rules have been constructed using XML, XSD and RDFS.

All of the above devised mechanisms and ontologies have been experimented with and evaluated using three datasets of events from three different pervasive systems. These three systems were chosen from three different environments (healthcare, e-commerce and production management). Results and outcomes are as expected and seem to be promising in the filed of Continuous Auditing and Continuous Monitoring. Our proposed methodologies can be used to audit and monitor Process based pervasive systems continuously where systems are not bound in one geographical location and has the facility of heterogeneous distributed data sources. These proposed methodologies can be adapted and commercialized directly to the today's emerging technologies after making respective changes for the specific industry domain.

In conclusion, we can now state that: a CPA system is one that detects and obtains evidence associated with a process for the purpose of augmenting

and assisting audit professionals in operational and compliance auditing in any EIS, and which invokes actions specified by and derived from policies and rules in a continuous manner. This dissertation research has produced a coherent framework design with implemented software modules that clearly demonstrate our successful outcomes.

Appendix A

: Meaning of Concepts and Propositions

In this appendix, we briefly introduce the main concepts and propositions in *Audit Rule Ontology*, which are necessary to help readers understand the methodologies presented in Chapters 7, 8 and 9.

Domain ontology consists of the technical terms that are used in the domain and the semantic relations among them. This is used to standardize the descriptions of domain knowledge, terms and relations that eliminate the dependency on domain experts.

Table A.1: Meaning of Concepts and Propositions

Concepts and/or Propositions	**Meaning or Definition**
Domain Ontology	consists of the technical terms that are used in the domain and the semantic relations among them. This is used to standardize the description of domain knowledge, terms and relations that eliminates the dependency on domain experts.
Audit Rule Ontology	consists of audit rules as primitive and semantic relations among them. It organize the *practical* and *verifiable* audit rules that either *directly* or *indirectly* enforceable.
Concept	is an unit of knowledge created by a unique combination of characteristics. How we think about things - what is meant by a word; what someone intends to express or what someone understands.
Proposition	the meaning of the statement.
Justify	is a relation between a stem (deep) rule and its branch (shallow) rule. It can be implemented using Deliberation RuleML under The Rule Markup Language which subsumes Datalog language via Higher Order Logic (HOL) and First Order Logic (FOL).

Continuation of Table A.1

Concepts and/or Propositions	**Meaning or Definition**
Depend-on	is a relation between a rule and another rule whose inference is essential condition for its inference.
Specialized	is a relation between a specialized rule and its class rule.
Override	override with explanation (Comment must be provided when the violation occurs). It is a relation between a specialized rule with some overrides and its class rule that it overrides.
Guideline	guideline (suggested, but not enforced).
Verify	is a global active Event-Condition-Action (ECA) reaction rule. It can be implemented using Reaction RuleML under The Rule Markup Language which is part of Descriptive Logic family.
Monitor	is a simple Event-Condition-Action (ECA) rule which is triggered by an event pattern in the on part. A matching event instance is asserted which triggers the pattern in the ECA rule. The ECA rule proofs the condition and executes the action which asserts a new fact to the knowledge base. This fact is then queried. This is Complex Event Processing (CEP).

Continuation of Table A.1	
Concepts and/or Propositions	**Meaning or Definition**
Rule	A rule can be optionally annotated with descriptive metadata (e.g. for life cycle management) using the meta role which is a single formula as descriptive meta-knowledge.
Rulebase	is a collection of rules that can be ordered or unordered, without or with duplicates.
Assert	is a performative/action wrapper specifying that its content is *asserted*, making an implicit assumption of Rulebase.
Implies	is an implication rule. It consists of a conclusion role 'then' followed by a premise role 'if', or, equivalently (since roles constitute unordered elements), a premise role followed by a conclusion role.
Retract	is a performative/action wrapper specifying that its content is to be *deleted*, making an 'implicit Rulebase' assumption
Consult	is a performative/action wrapper that dynamically "consults" (imports) the knowledge form an 'enclosing' 'Message' or an external knowledge source.

Continuation of Table A.1	
Concepts and/or Propositions	**Meaning or Definition**
Update	is an action which executes an update of the knowledge base. Update actions can be used as performatives and as complex actions in the 'do' role of a reaction rule.
Entails	Used to 'Assert'/'Query' that/whether the sequence of formulas in the first 'Rulebase' entails the sequence of formulas in the second, e.g. the first acting as a knowledge base and the second acting as its integrity constraints.
Atom	a logical atom, i.e. an expression formed from a predicate (or relation) applied to a collection of operands, or a frame object.
Message	is an element that provides the syntax for inbound and outbound messages / notifications.
Receive	is an performative/action that waits to receive an 'enclosed' 'Message' matching the pattern defined in the Message's signature definition.
Send	is an performative/action that sends an 'enclosed' 'Message'. The Send action is used in messaging reaction rules in CEP Reaction RuleML in the action part.

Continuation of Table A.1	
Concepts and/or Propositions	**Meaning or Definition**
Action	Explicit generic Action. An action can be defined with positional arguments, with unpositional slots, by an action expression function, as a complex action by an action algebra operator, as frame object, and by external/internal reference attributes.
Query	an performative/action wrapper specifying that its content is *queried*, making an 'implicit 'Rulebase' assumption
Answer	is a performative wrapper giving the answers to a 'Query' or results from a forward directed rule processing/reasoning as in the case of production rules and reaction rules with actions.
Enclosed	is a role enclosing a RuleML 'Message' in messaging actions/primitives ('Consult' 'Send' 'Receive').

Appendix B

: Figures and Charts

In this appendix, we listed all the figures and charts that are too large to fit in one page and has used in many Chapters to explain concepts and to test hypotheses, especially in evaluation Chapter 9 in experimental settings Section 9.4.

B.1 Process Map and Description of Healthcare Decision Support System Processes

A complete process map of cervical cancer patient in Gynaecology department of Dutch Academic Hospital [75] is illustrated in Figure B.1. Each process (case) is a patient of a Gynaecology department. The event log captures treatment procedures pertaining to 11 different diagnosis codes described in Appendix C. Various diversity of process instances were in the dataset in the event log with data attributes related to diagnosis and treatment. We tested

our hypotheses on Diagnosis code and treatment code sub-processes in Section 9.4.2 in Chapter 9. Process map (Figure B.1) is expanded in Figure B.2.

B.2 Process Map and Description of E-commerce System Processes

A complete process of loan application process in E-commerce Management System [76] of Dutch Financial Institute is illustrated in Figure B.3. Each process represented in the event log is an application process for a personal loan or overdraft within a global financing organization. The event log contains events from three intertwined subprocesses, which can be distinguished by the first letter of each event name (**A**, **O** and **W**). The **A** subprocess is concerned with handling the applications themselves. The **O** subprocess handles offers send to customers for certain applications. The **W** process describes how work items, belonging to the application, are processed. **A** and **O** are considered as operational and **W** considered as compliance subprocesses. We tested our hypotheses on all three subprocesses as individual processes in Section 9.4.2 in Chapter 9. Process map (Figure B.3) is expanded in Figure B.4.

B.3 Process Map and Description of Production Monitoring System Processes

An incident and problem handling system of Volvo IT which is called VINST [70] that mainly supports the following two types of processes: (a) handle

incidents process and (b) handle activity process.

Handle Incidents Problem Process restores normal service operation (Normal service operation' that is defined within Service Level Agreement (SLA)) as quickly as possible and by that ensuring the best possible levels of service quality and availability are maintained. Incidents that cannot be resolved by the Service Desk or Expert Helpdesk should be escalated to Second Line and/or Third Line teams. Solution should be established as quickly as possible in order to restore the service to normal with minimum disruption to the business. After implementing a Solution by IT departments (and specialist teams) and *verifying that the service is restored* the Incident is closed. If the Action Owner suspects that the Incident might reoccur a *Problem record shall be registered*.

Handle Activity Problem Process diagnoses the root cause(s) incidents activities and secures the resolution of those problems to enhance the quality of IT-services delivered and/or operated by Volvo IT. Handle Activity Problem Process works together with other processes like Handle Incidents Monitor service, Discover & Define Opportunity, Develop, Deploy & Provide and Manage Service Change etc. to ensure that IT service availability and quality are increased. Handle Activity Problem Process also, when applicable, *verify and update Solutions in the knowledgebase*, so that the best possible Solution is available during the life-cycle of the problem.

Complete process maps of both Incidents and Activity problem sub-processes are illustrated in Figures B.5 and B.6, respectively.

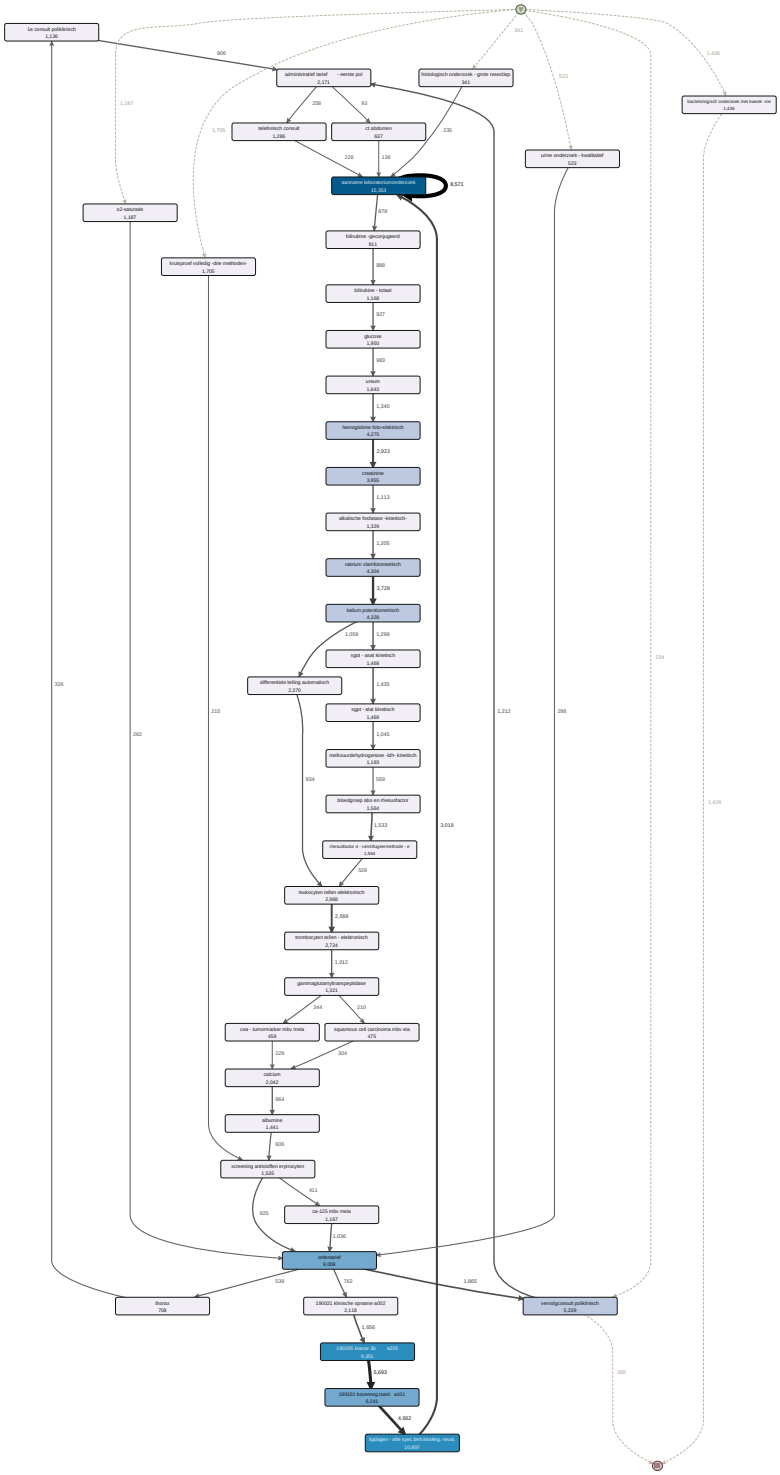
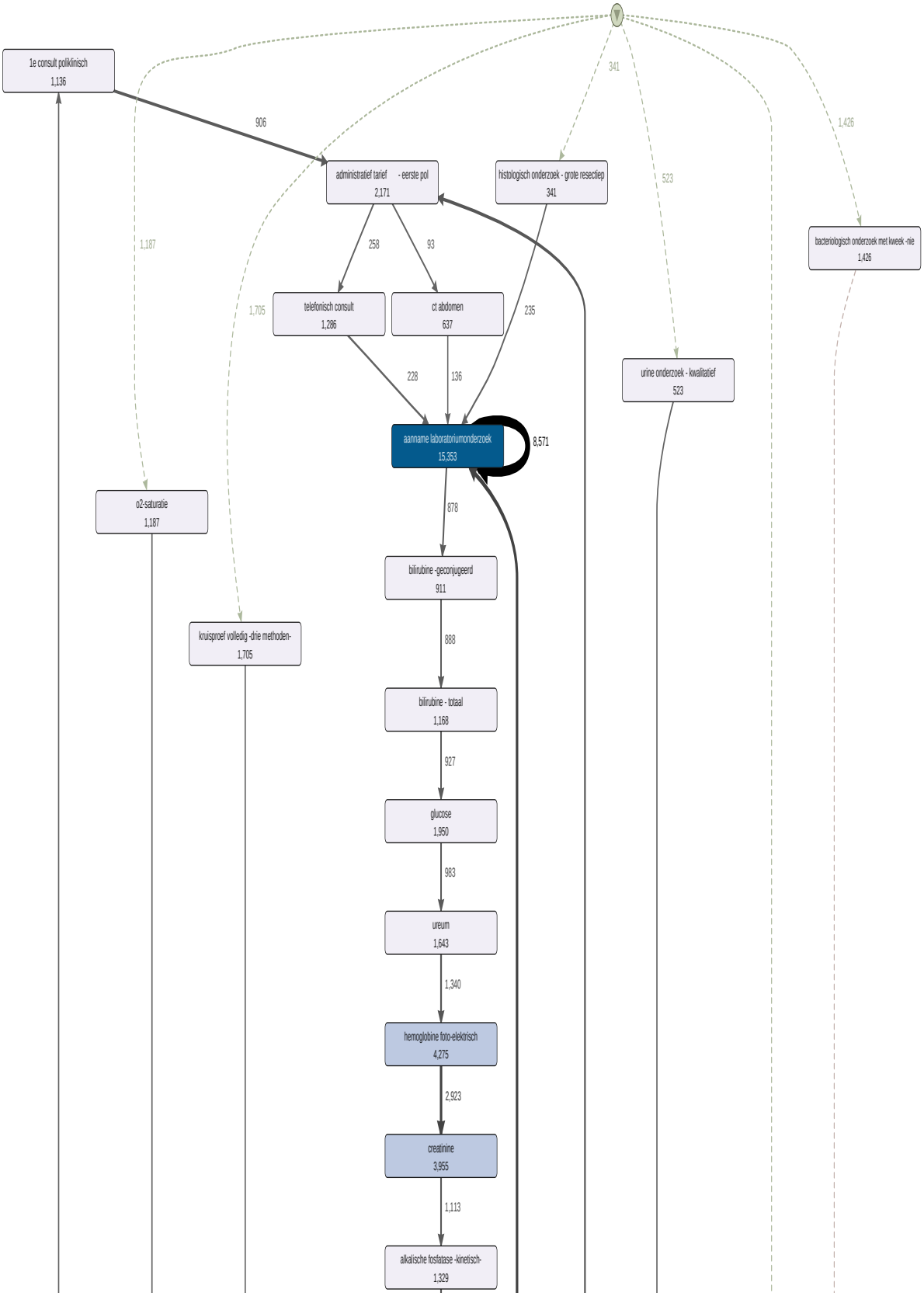
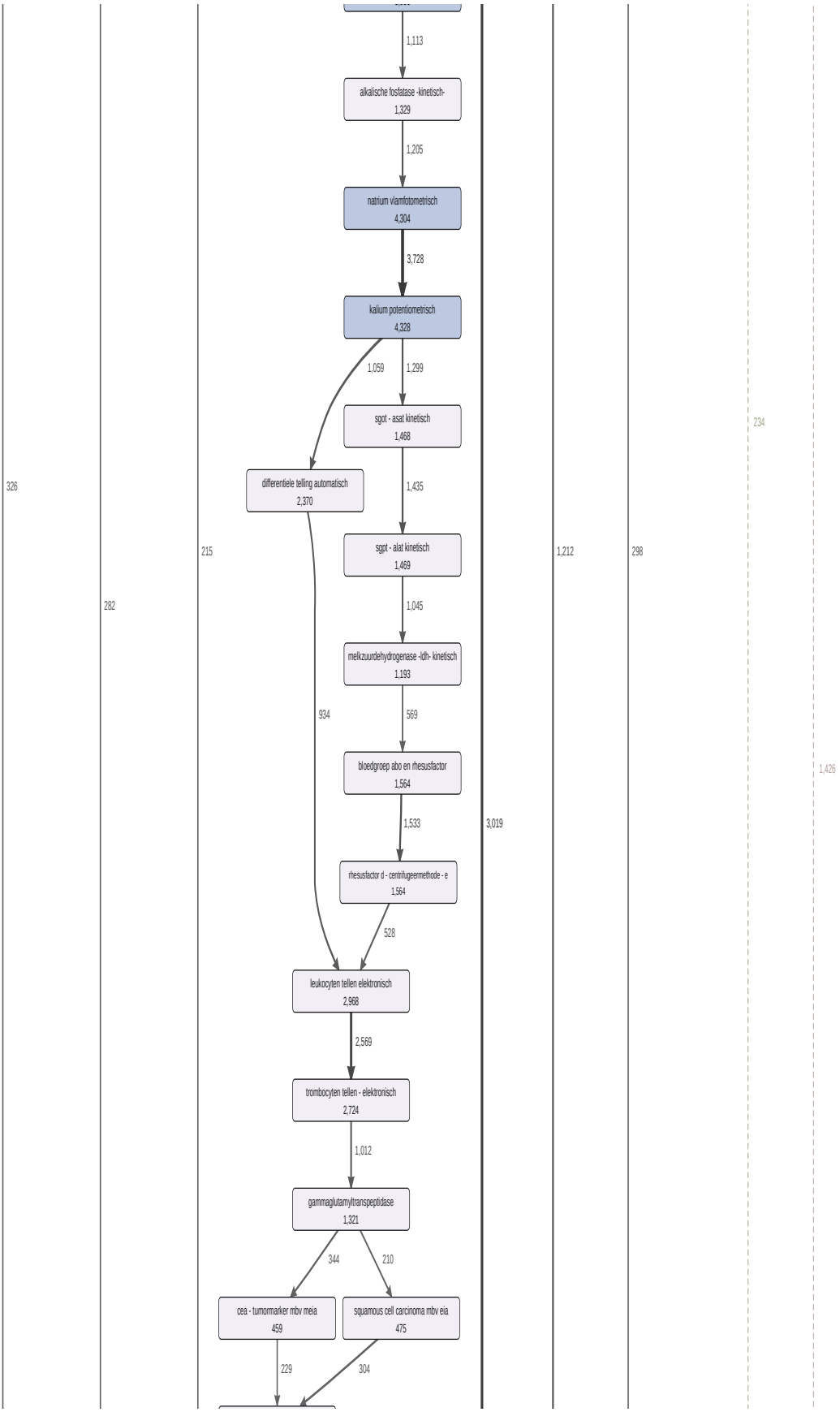


Figure B.1: Dutch Academic Hospital - *Process Map* of cervical cancer patient in Gynaecology department [75]


1e consult polikliniisch
1,136
906
administratief tarief - eerste pol
2,171
histologisch onderzoek - grote resectiep
341
341
523
1,426
bacteriologisch onderzoek met kweek -nie
1,426
1,187
1,705
258
93
telefonisch consult
1,286
ct abdomen
637
235
228
136
urine onderzoek - kwalitatief
523
aanname laboratoriumonderzoek
15,353
8,571
o2-saturatie
1,187
878
bilirubine -geconjugeerd
911
kruisproef volledig -drie methoden-
1,705
888
bilirubine - totaal
1,168
927
glucose
1,950
983
ureum
1,643
1,340
hemoglobine foto-elektrisch
4,275
2,923
creatinine
3,955
1,113
alkalische fosfatase -kinetisch-
1,329


1,113
alkalische fosfatase -kinetisch-
1,329
1,205
natrium vlamfotometrisch
4,304
3,728
kalium potentiometrisch
4,328
1,059
1,299
sgot - asat kinetisch
1,468
differentiele telling automatisch
2,370
1,435
sgot - alat kinetisch
1,469
1,045
melkzuurdehydrogenase -ldh- kinetisch
1,193
934
569
bloedgroep abo en rhesusfactor
1,564
1,533
rhesusfactor d - centrifugeermethode - e
1,564
528
leukocyten tellen elektronisch
2,968
2,569
trombocyten tellen - elektronisch
2,724
1,012
gammaglutamyltranspeptidase
1,321
344
210
cea - tumormarker mbv meia
459
squamous cell carcinoma mbv eia
475
229
304
326
282
215
3,019
1,212
298
234
1,426

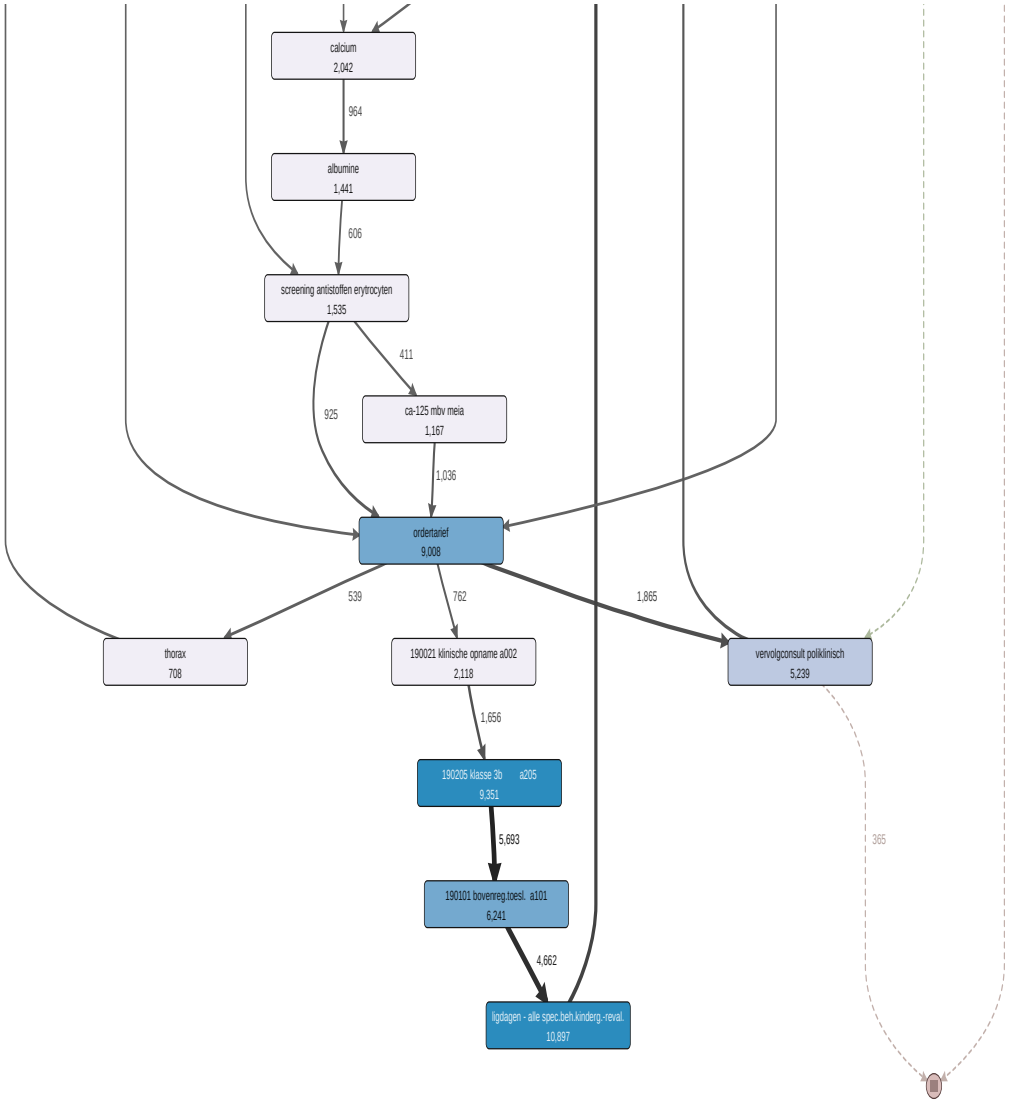


Figure B.2: Dutch Academic Hospital - *Process Map* of cervical cancer patient in Gynaecology department [75] (expanded)

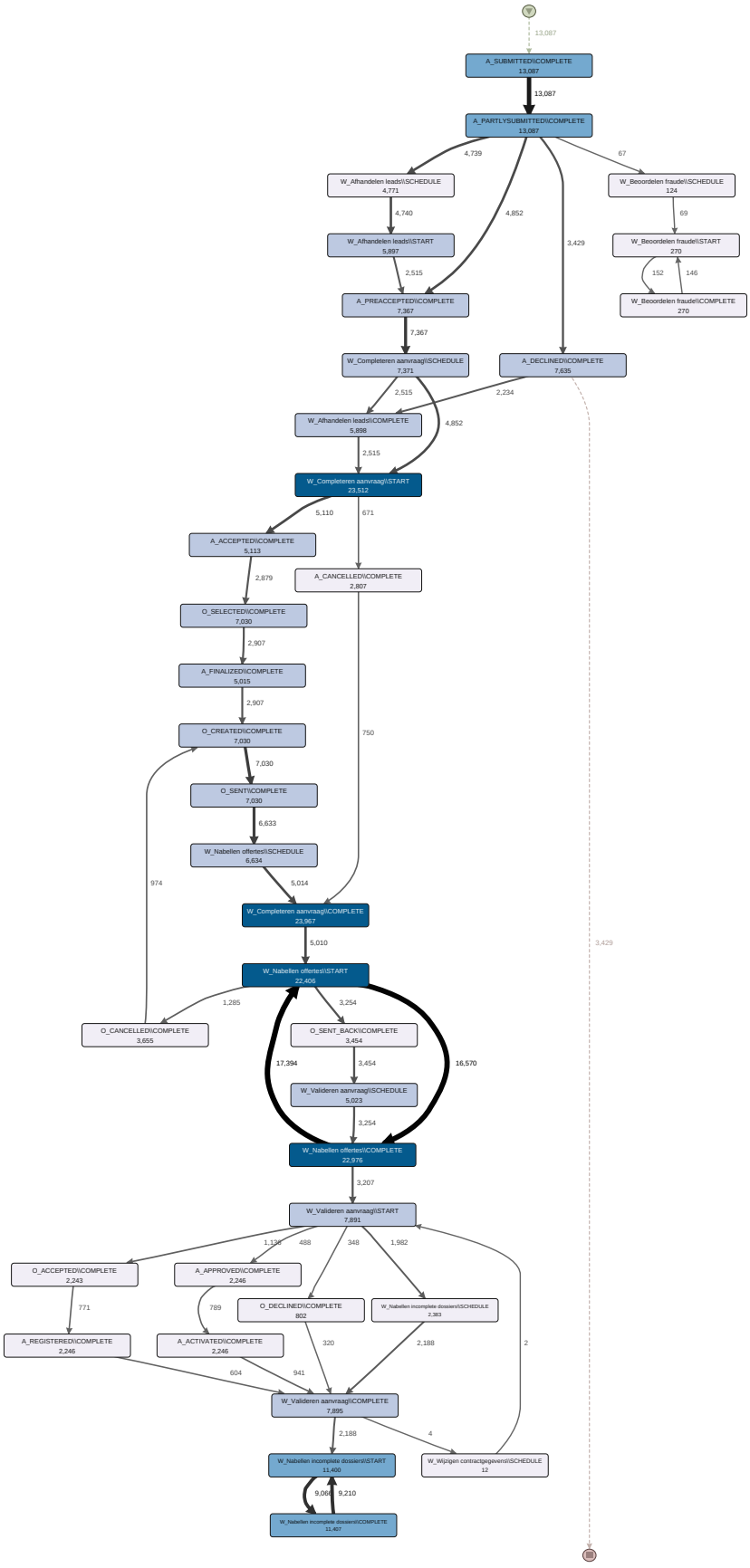
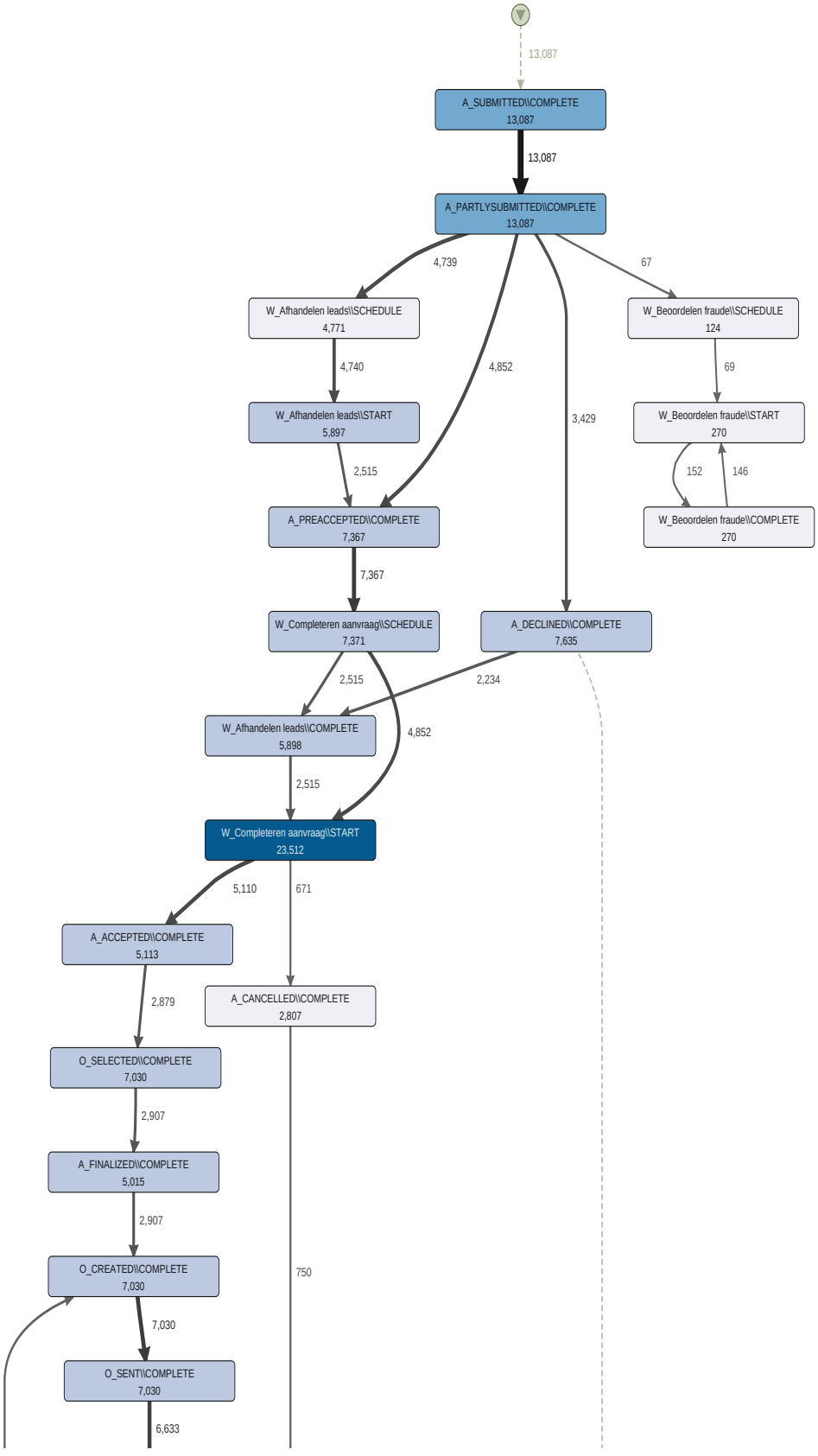
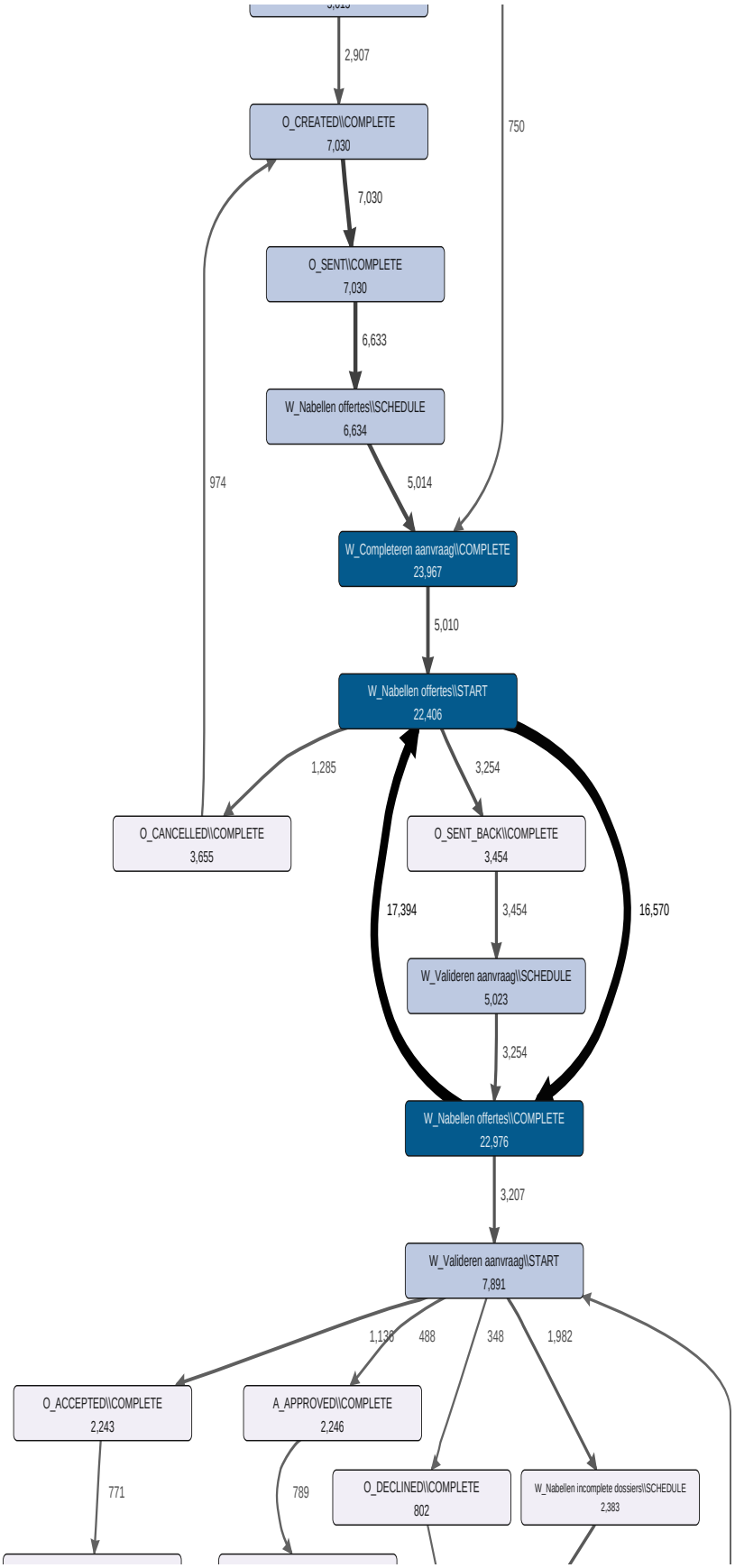


Figure B.3: Dutch Financial Institute - *Process Map* of application for personal loan or overdraft [76]


13,087
A_SUBMITTED\\COMPLETE
13,087
13,087
A_PARTLYSUBMITTED\\COMPLETE
13,087
4,739
4,852
3,429
67
W_Afhandelen leads\\SCHEDULE
4,771
W_Beoordelen fraude\\SCHEDULE
124
4,740
69
W_Afhandelen leads\\START
5,897
W_Beoordelen fraude\\START
270
2,515
152
146
A_PREACCEPTED\\COMPLETE
7,367
W_Beoordelen fraude\\COMPLETE
270
7,367
W_Completeren aanvraag\\SCHEDULE
7,371
A_DECLINED\\COMPLETE
7,635
2,515
2,234
W_Afhandelen leads\\COMPLETE
5,898
4,852
2,515
W_Completeren aanvraag\\START
23,512
5,110
671
A_ACCEPTED\\COMPLETE
5,113
2,879
A_CANCELLED\\COMPLETE
2,807
O_SELECTED\\COMPLETE
7,030
2,907
A_FINALIZED\\COMPLETE
5,015
2,907
750
O_CREATED\\COMPLETE
7,030
7,030
O_SENT\\COMPLETE
7,030
6,633


2,907
O_CREATED\\COMPLETE
7,030
750
7,030
O_SENT\\COMPLETE
7,030
6,633
W_Nabellen offertes\\SCHEDULE
6,634
974
5,014
W_Completeren aanvraag\\COMPLETE
23,967
5,010
W_Nabellen offertes\\START
22,406
1,285
3,254
O_CANCELLED\\COMPLETE
3,655
O_SENT_BACK\\COMPLETE
3,454
17,394
3,454
16,570
W_Valideren aanvraag\\SCHEDULE
5,023
3,254
W_Nabellen offertes\\COMPLETE
22,976
3,207
W_Valideren aanvraag\\START
7,891
1,136
488
348
1,982
O_ACCEPTED\\COMPLETE
2,243
A_APPROVED\\COMPLETE
2,246
771
789
O_DECLINED\\COMPLETE
802
W_Nabellen incomplete dossiers\\SCHEDULE
2,383

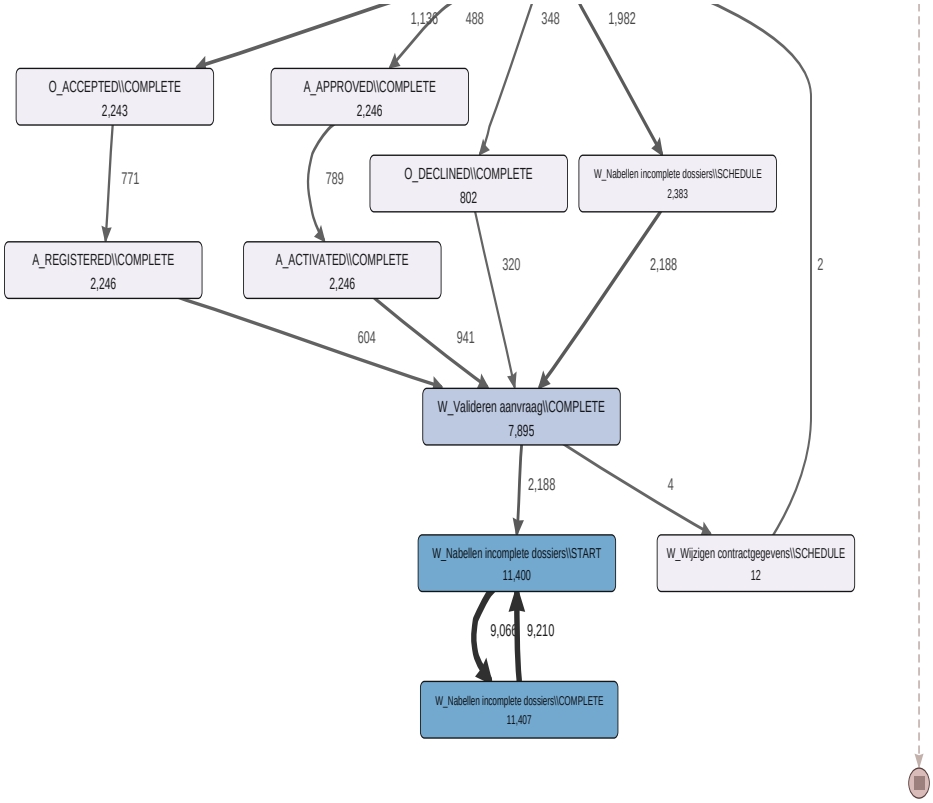


Figure B.4: Dutch Financial Institute - *Process Map* of application for personal loan or overdraft [76] (expanded)

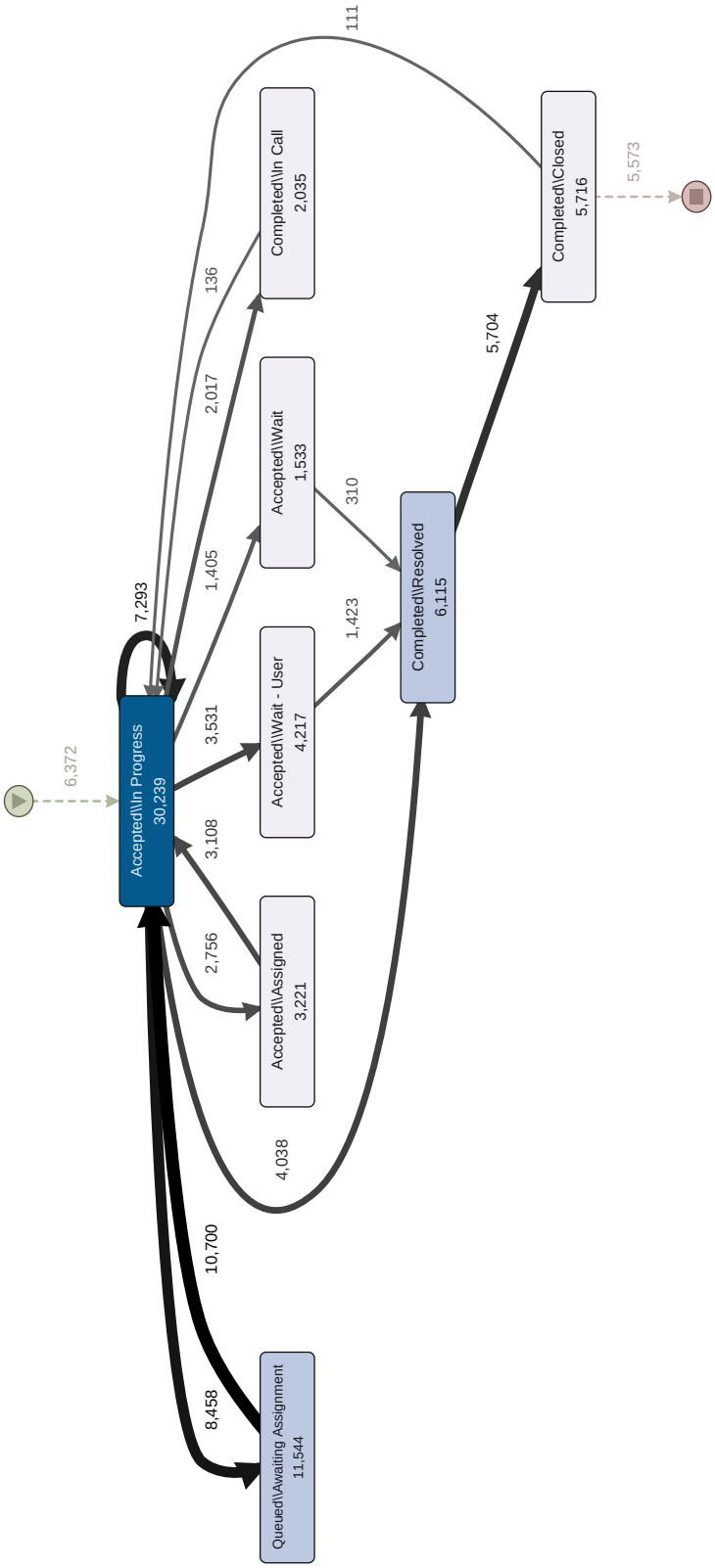


Figure B.5: Volvo IT Belgium - *Process Map* of VINST incidents management sub-system [70]

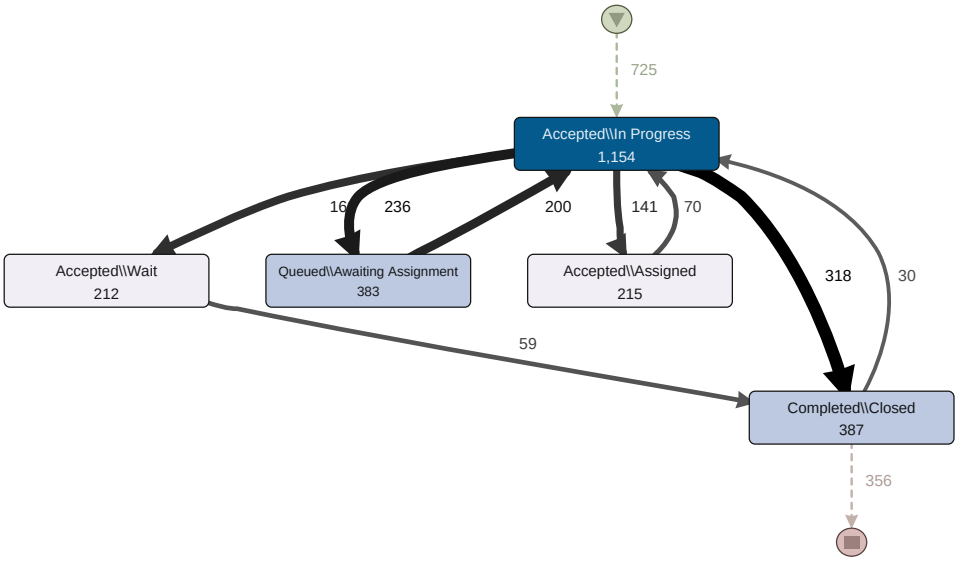


Figure B.6: Volvo IT Belgium - *Process Map* of VINST open problems management sub-system [70]

Appendix C

: Description of Diagnosis of Patients

We have used the HDSS dataset to test our hypothesis of PO, AROP and CPA in Section 9.2.1 of evaluation Chapter 9. We have tested our hypotheses on several 'diagnosis code' and 'treatment code'. Following audit rules has experimented on the stages of malignancy of cervical cancer in diagnosis code (M13) and treatment code 803.

Operational Audit Rules: "**Verify** the stages of (malignancy) cervical cancer in diagnosis code (M13)".

Compliance Audit Rules: "**Justify** the urgency of an activity: g0 (hemoglobin photoelectric)" that diagnosed with M13 and whose treatment code is 803.

The HDSS dataset is an anonymized event log of a Dutch Academic Hospital [75]. Each process (case) is a patient of a Gynaecology department. The event log captures treatment procedures pertaining to 11 different diagnosis codes described below. The stage of malignancy is not known for some cases

in all the diagnosis categories.

Table C.1: Diagnosis Code and Description of Diagnosis of Patients.

Diagnosis Code	Diagnosis Description
M11	Pertains to the cancer of the vulva that describes the information on cases diagnosed with squamous cell carcinoma (stages I, II, III1, III2, IVa and IVb), malignant neoplasms and melanoma, basal cell carcinoma, borderline malignancy
M12	Pertains to the cancer of the vagina that describes the information on cases diagnosed with squamous cell carcinoma (stages II, III and IVb), malignant neoplasms, adenocarcinoma (stage II). Certain metastases cases are also included.
M13	Pertains to the cancer of the cervix (uteri) that describes information on cases diagnosed with squamous cell carcinoma (stages Ia1, Ia2, Ib, IIa, IIb, IIIb, IVa and IVb), malignant neoplasms, adenocarcinoma (stages Ia1, Ib and IIa), borderline malignancy, sarcoma

Continuation of Table C.1

Diagnosis Code	Diagnosis Description
M14	Pertains to the cancer of the corpus uteri. Describes information on cases diagnosed with adenocarcinoma (stages Ia, Ib, Ic, IIa, IIb, IIIa, IIIb, IVa and IVb), malignant neoplasms and endometrium, clear cell carcinoma (stages Ib and IIIb), borderline malignancy. Certain metastases cases are also included.
M15	Primarily pertains to the cancer of the corpus uteri of type sarcoma (stages II and III according to the FIGO staging system). However, certain cases of colon cancer and myometrium are also classified into this category
M16	Pertains to the cancer of the ovary that describes information on cases diagnosed with adenocarcinoma of types • serous (stages Ia, Ic, IIa, IIIb, IIIc and IV) • endometroid (stages Ic, IIIc) • mucinous (stages Ic, IIc and IIIc) • non-differentiated (stages IIIc and IV) non-epithelial malignancy (stages Ia, IIa, IIIa and IIIc), neoplasms, borderline malignancy, clear cell carcinoma. Certain metastases cases are also included.

Continuation of Table C.1

Diagnosis Code	Diagnosis Description
821	Pertains to the cancer of the ovary that describes information on cases diagnosed with adenocarcinoma of types • serous (stage IIIc) • mucinous (stage IIIc) non-epithelial malignancy, neoplasms
822	Pertains to the cancer of the cervix (uteri) that describes information on cases diagnosed with squamous cell carcinoma (stage Ib), adenocarcinoma (stages IIa and IIb), borderline malignancy, malignant neoplasms
106	Describes a heterogeneous mix of cases pertaining to the cancers of cervix uteri - of types squamous cell carcinoma (stages Ia and IIa), malignant neoplasms and borderline malignancy; vulva - of types squamous cell carcinoma (stages III2, IVa and IVb) and malignant melanoma; corpus uteri - of types adenocarcinoma (stages Ib, Ic and IIa), malignant neoplasms and borderline malignancy; vagina - endometrium and ovarian tube

Continuation of Table C.1

Diagnosis Code	Diagnosis Description
823	Describes a heterogeneous mix of cases pertaining to the cancers of corpus uteri - of types adenocarcinoma (stages IVa and IVb), malignant neoplasms and sarcoma (stage IVb according to the FIGO staging system); ovary - of type serous adenocarcinoma (stage IIIc); endometrium
839	Describes a heterogeneous mix of cases pertaining to the cancers of ovary - of types serous adenocarcinoma (stages IIIc and IV) and borderline malignany; uterine appendages - of type malignant neoplasms; vulva - of type malignant neoplasms

Appendix D

: Tools and Technologies

In this appendix, we list all tools and technologies that used in Chapter 9 for experiment designing and experimental settings to test our hypotheses and to present our interpretation of results.

Disco

Disco[a] is a process mining technology that helps to create visual maps and actionable insights from process data. It also has the capabilities to optimize performance, to control deviations of processes & sub-processes, and to explore variations of different processes & sub-processes using various constraints and rules.

In Chapter 9, we use **Disco** extensively to explore and to mine processes and sub-processes in all three datasets [75, 76, 70]. All the visual maps and figures (in Chapter 9 and in Appendix B) of processes and sub-processes were made possible by using **Disco**'s visualization and mapping feature.

[a] http://fluxicon.com/disco

University of Windsor become academic partner with Fluxicon's Academic Initiative for Process Mining Research (http://fluxicon.com/academic/) and we obtained free academic license on Jan 14, 2016 to use **Disco** for process mining research and education purposes only.

ProM 6

ProM 6[b], a process mining tool, is distributed as a downloadable package using the GNU Public License (GPL) open source license. **ProM 6** plug-ins are distributed as separate packages under GPL and it also support the import of (and the conversion between) several process modelling languages, such as: Petri nets [c] (PNML, TPN), EPCs / EPKs [d] (Aris graph format, EPML), YAWL[e]. There are more than 230 plug-ins available which includes supporting control-flow mining techniques (such as the Alpha algorithm, Genetic mining, Multi-phase mining etc.), different kind heuristics miner for flexible processes, decision miner for data perspective mining as well as network or assignment based mining for organizational perspective mining.

In Chapter 9, we use **ProM 6** and several heuristic mining plug-ins to mine Dutch Academic Hospital's[75] cervical cancer patients in Gynaecology department as well as loan approval process in Dutch Financial Institute[76] datasets.

[b] http://www.promtools.org
[c] http://www.informatik.uni-hamburg.de/TGI/PetriNets/
[d] http://www.epk-community.de/
[e] http://www.yawl-system.com/

Protégé

An open source, free platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. **Protégé** is supported by a strong community of academic, government, and corporate users, who use **Protégé** to build knowledge-based solutions in areas as diverse as biomedicine, e-commerce, and organizational modeling. **Protégé** is freely available to download at http://protege.stanford.edu/ and wiki is http://protegewiki.stanford.edu/wiki/Main_Page.

In Chapter 9, we use **Protégé** to develop all three kind of ontologies as well as mapping ontology to build Hybrid Layered Ontology for each datasets. We use SPARQL to query onto ontology that takes the description of what the application wants, in the form of a query, and returns that information, in the form of a set of bindings or an RDF graph, including OWL reasoning.

Prova

An open source rule language for reactive agents and event processing rules engine. It combines imperative, declarative and functional programming styles. **Prova**[f] is a highly expressive distributed rule engine that supports complex reaction rule-based workflows, rule-based complex event processing, distributed inference services, rule interchange, rule-based decision logic and dynamic access to external data sources, web-based services and Java APIs.

Prova follows the spirit and design of the recent W3C Semantic Web initiative and combines declarative rules, ontologies and inference with dynamic

[f] https://prova.ws/index.html

object-oriented programming and access to external data sources. One of the key advantages of Prova is its elegant separation of logic, data access, and computation and its tight integration of Java, Semantic Web technologies, enterprise service-oriented computing and complex event processing technologies.

In Chapter 9, we use **Prova** as a rule engine for processing and inferring audit rules to enforce actions on the detected events. We also construct audit rules using **Prova**'s rule language for reactive agents and event processing.

JESS

JESS is a rule engine and scripting environment written entirely in Oracles's® Java™ language by Ernest Friedman-Hill at Sandia National Laboratories[g] in Livermore, CA. **JESS** uses an enhanced version of the Rete algorithm to process rules. Rete is a very efficient mechanism for solving the difficult many-to-many matching problem [30]. **JESS** has many unique features including backwards chaining and working memory queries, and of course **JESS** can directly manipulate and reason about Java objects.

In Chapter 8, we used **JESS** in the Ontology evolution of the addition, suppression and modification operations on the knowledge base using the **JESS** language "assert" for addition, "modify" for modification, "retract" for suppression. The **JESS** facts are of triplets type given by (Predicate, Subject, Object). It is compulsory that the development of a system which propagates the changes automatically.

[g] http://www.jessrules.com/jess/index.shtml

Academic use license (Ref# 10102) was obtained from Sandia National Laboratories[h] for the United States Department of Energy on Aug 24, 2015 to use **JESS** for Non Proprietary Research and Development purposes only at the University of Windsor.

[h] http://www.sandia.gov/

Bibliography

[1] Wil Aalst, Kees Hee, Jan Werf, Akhil Kumar, and Marc Verdonk. Toward risk assessment as a service in cloud environments. *HotCloud*, pages 1–7, 2010. [cited at p. 29]

[2] Wil Aalst, Kees Hee, Jan Werf, Akhil Kumar, and Marc Verdonk. Conceptual model for online auditing. *Decision Support Systems*, 50:636–647, 2011. [cited at p. 30]

[3] S. Achour, M. Dojat, C. Rieux, P. Bierling, and E. Lepage. A umls-based knowledge acquisition tool for rule-based clinical decision support system development. *Journal of the American Medical Informatics Association*, 8(4):351–360, 2001. [cited at p. 79]

[4] M. Alles, G. Brennan, A. Kogan, and M. Vasarhelyi. Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at siemens. *International Journal of Accounting Information Systems*, 7(2):137–161, 2006. [cited at p. 29, 79]

[5] M. Alles, A. Kogan, and M. Vasarhelyi. Feasibility and economics of continuous assurance. *Auditing: A Journal of Practice & Theory*, 21(1):1–14,

2002. [cited at p. 21, 37]

[6] M. Alles, A. Kogan, and M. Vasarhelyi. Real time reporting and assurance: Have their time come? *Institute of Chartered Financial Analysis of India*, Special Issue(Finance), 2004. [cited at p. 15, 19, 28]

[7] M. Alles, A. Kogan, and M. Vasarhelyi. Restoring auditor credibility - tertiary monitoring and logging of continuous assurance systems. *Int'l Journal of Accounting Info. Sys.*, 5:183–202, 2004. [cited at p. 37, 39, 43]

[8] M. Alles, A. Kogan, and M. Vasarhelyi. Audit automation for implementing continuous auditing: Principles and problems. *Rutgers Business School*, pages 1–24, 2008. [cited at p. 19, 21]

[9] M. Alles, A. Kogan, and M. Vasarhelyi. Putting continuous auditing theory into practice: Lessons from two pilot implementations. *Journal of Information Systems*, 22(2):195–214, 2008. [cited at p. 31]

[10] M. Alles, A. Kogan, M. Vasarhelyi, and J. Wu. Continuity equations: Analytical monitoring of business processes in continuous auditing. *12th World Continuous Auditing Symposium*, 2006. [cited at p. 30, 45, 79]

[11] M. Alles, F. Tostes, M. Vasarhelyi, and E. Riccio. Continuous auditing - the usa experience and considerations for its implementation in brazil. *Journal of Information Systems and Technology Management*, 3(2):211–224, 2006. [cited at p. 22]

[12] D. Arter. Process-based auditing. *American Society for Quality - Columbia Basin*, June, 2007. [cited at p. 30]

[13] R. Baksa and M. Turoff. Continuous auditing as a foundation for real time decision support - implementation challenges and successes. *Supporting Real Time Decision-Making, Annals of Info. Sys.*, 13:237–252, 2011. [cited at p. 3, 59]

[14] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May, 2001. [cited at p. 3, 59]

[15] S. Bhattacharya, D. Xu, and K. Kumar. An ANN-based Auditor Decision Support System Using Benford's Law. *Decision Support Systems*, 50(3):576–584, 2011. [cited at p. 46]

[16] M. Bovee, A. Kogan, K. Nelson, R. Srivastava, and M. Vasarhelyi. Financial reporting and auditing agent with net knowledge (FRAANK) and extensible business reporting language (XBRL). *Journal of Information Systems*, 19(1):19–41, 2005. [cited at p. 22]

[17] C. Brown, J. Wong, and A. Baldwin. Research streams in continuous audit: A review and analysis of the existing literature. *12th Continuous Auditing and Reporting Symposium*, pages 1–13, 2006. [cited at p. 21]

[18] Organizing business knowledge: the MIT process handbook. *Malone, T. and Crowston, K. and Herman, G.* The MIT Press, 2003. [cited at p. 62]

[19] P. Byrnes, A. Al-Awadhi, B. Gullvist, H. Brown-Liburd, R. Teeter, D. Warren Jr., and M. Vasarhelyi. Evolution of auditing - from the traditional approach to the future audit. *AICPA -White Paper*, November:1–9, 2012. [cited at p. 22, 29, 150]

[20] P Byrnes, B. Ames, M. Vasarhelyi, and D. Warren Jr. The current state of continuous auditing and continuous monitoring. *AICPA -White Paper*, October:1–15, 2012. [cited at p. 22, 29]

[21] N. Casellas. Chapter 3: Methodologies, tools and languages for ontology design. *Legal Ontology Engineering*, Law, Governance and Technology Series(3):57–107, 2011. [cited at p. 86]

[22] CaseWare. Continuos auditing: A strategic approach to implementation, 2008. [cited at p. 29]

[23] S. Casteleyn, P. Plessers, and O. De Troyer. On generating content and structural annotated websites using conceptual modeling. *Conceptual Modeling (ER2006)*, LNCS 4215:267–280, 2006. [cited at p. 82]

[24] D. Chan and M. Vasarhelyi. Innovation and practice of continuous auditing. *Int'l Journal of Accounting Info. Sys.*, 12:152–160, 2011. [cited at p. 21, 29, 78]

[25] R. Dull, D. Tegarden, and L. Schleifer. Actve: A proposal for an automated continuous transaction verification environment. *Journal of Emerging Technologies in Accounting*, 3:81–96, 2006. [cited at p. 21]

[26] C. Durtschi, W. Hillison, and C. Pacini. The effective use of benford's law to assist in detecting fraud in accounting data. *Journal of Forensic Accounting*, V:17–34, 2004. [cited at p. 46]

[27] R. Elliott. Twenty-first century assurance. *Auditing: A Journal of Practice and Theory*, 21(1):139–146, 2002. [cited at p. 21, 37]

[28] S. Flowerday, A. Blundell, and R. Solms. Continuous auditing technologies and models: A discussion. *Computer & Security*, 25:325–331, 2006. [cited at p. 21]

[29] S. Flowerday and R. Solms. Continuous auditing: verifying information integrity and providing assurances for financial reports. *Computer Fraud & Security*, July:12–16, 2005. [cited at p. 19]

[30] C. Forgy. Rete: A fast algorithm for the many pattern - many object pattern match problem. *Artificial Intelligence*, 19(1):17–37, 1982. [cited at p. 180]

[31] E. Friedman-Hill. *JESS in action : Rule-Based system in Java*. Manning Publications, isbn:1930110898 edition, 2003. [cited at p. 60, 81]

[32] F. Giorgos, P. Dimitris, and A. Grigoris. Evolving ontology evolution. *Current Trends in Theory and Practice of Computer science*, SOFSEM 06, 2006. [cited at p. 87]

[33] S. Groomer and U. Murthy. Continuous auditing of database applications: An embedded audit module approach. *Journal of Info. Sys.*, 3(2):53–69, 1989. [cited at p. 21]

[34] Object Management Group. *Semantics of Business Vocabulary and Business Rules (v 1.2)*, 2013. [cited at p. 70]

[35] The AICPA Assurance Services Executive Committee (ASEC) Emerging Assurance Technologies Task Force Continuous Assurance Working

Group. Audit analytics and continuous audit - looking toward the future, 2015. [cited at p. 32]

[36] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. [cited at p. 2, 59, 60, 74]

[37] F. Harmelen and D. Fensel. Practical knowledge representation for the web. *Proc. 16th Int'l Joint Conf. Artificial Intelligence*, 1999. [cited at p. 3, 59]

[38] I. Horrocks. DAML+OIL: A Description Logic for the Semantic Web. *IEEE Data Enggineering*, 25(1):4–9, 2002. [cited at p. 3, 59]

[39] S. Huang, D. Yen, L. Yang, and J. Hua. An investigation of Zipf's Law for fraud detection. *Decision Support Systems*, 46:70–83, 2008. [cited at p. 48]

[40] M. Jans, M. Alles, and M. Vasarhelyi. Process mining of event logs in auditing - opportunities and challenges. *Rutgers Business School*, pages 1–32, 2010. [cited at p. 90, 141]

[41] M. Jans, M. Alles, and M. Vasarhelyi. The case for process mining in auditing - sources of value added and areas of application. *International Journal of Accounting Information Systems*, 14:1–20, 2013. [cited at p. 90, 141]

[42] S. Jean, G. Pierra, and Y. Ait Ameur. A domain ontologies : A database-oriented analysis. *LNBIP: Web Info Systems and Technologies*, pages 238–254, 2007. [cited at p. 74, 79]

[43] A. Kanellou and C. Spathis. Auditing in enterprise system environment: a synthesis. *Journal of Enterprise Information Management*, 24(6):494–519, 2011. [cited at p. 78]

[44] S. Kemsley. Business process discovery. *TIBCO Software Business Report*, pages 1–12, 2011. [cited at p. 141]

[45] R. D. Kent, A. Zahid, and A. Snowdon. Continuous Auditing for Health Care Decision Support Systems. *Intelligent Decision Technologies*, SIST(10):731–741, 2011. [cited at p. 78]

[46] M. Klein. Combining and relating ontologies: an analysis of problems and solutions. *Proceedings - IJCAI Workshop*, pages 53–62, 2001. [cited at p. 4, 59]

[47] A. Kogan, M. Alles, M. Vasarhelyi, and J. Wu. Analytical procedures for continuous data level auditing: Continuity equations. *Rutgers Business School*, pages 1–49, 2010. [cited at p. 28, 45]

[48] A. Kogan, E. Sudit, and M. Vasarhelyi. Continuous online auditing: A program of research. *Journal of Info. Sys.*, 13(2):87–103, 1999. [cited at p. 19, 20, 21]

[49] J. Kuhn and S. Sutton. Continuous auditing in erp system environments: The current state and future directions. *Journal of Info. Sys.*, 24(1):91–112, 2010. [cited at p. 21]

[50] K. Kumar and J. Hillegersberg. Enterprise resource planning experiences and evolution. *Common ACM*, 43(3):22–26, 2000. [cited at p. 78]

[51] S. Li, S. Huang, and Y. Lin. Developing a continuous auditing assistance system based on information process models. *Journal of Computer Information Systems*, Fall(2007):1–13, 2007. [cited at p. 21]

[52] S. Lipovetsky. Pareto 80-20 law - derivation via random partitioning. *International Journal of Mathematical Education in Science and Technology*, 40(2):271–277, 2009. [cited at p. 49]

[53] D. McGuinness and F. Harmelen. OWL Web Ontology Language overview. Technical report, 2004-2009. [cited at p. 79]

[54] J. Mei, E. Bontas, and Z. Lin. OWL2Jess: A transformational implementation of the OWL Semantics. *Parallel and Distributed Processing and Applications-ISPA 2005 Workshops*, LNCS 3759:599–608, 2005. [cited at p. 81, 87]

[55] U. Murthy and S. Groomer. A continuous auditing web services (CAWS) model for XML based accounting systems. *International Journal of Accounting Information Systems*, 5:139–163, 2004. [cited at p. 40, 42]

[56] M. Nigrini. Continuous auditing research report. Technical report, Canadian Institute of Chartered Accountants (CICA) and American Institute of Certified Public Accountants (AICPA), 1999. [cited at p. 18]

[57] M. Nigrini. Continuous auditing. *Ernst & Young Center for Auditing Research and Advanced Technology*, University of Kansas, 2000. [cited at p. 18]

[58] C. Nyulas, M. O'Connor, and T. Samson. Datamaster - a plug-in for importing schemas and data from relational databases into protg stanford

medical informatics. *Stanford University School of Medicine, Stanford, CA 94305*, 2007. [cited at p. 80]

[59] K. Omoteso, A. Patel, and P. Scott. Information and communications technology and auditing: Current implications and future directions. *International Journal of Auditing*, 14:147–162, 2010. [cited at p. 21]

[60] R. Onions. Towards a paradigm for continuous auditing. *http://www.auditsoftware.net/community/how/run/tools/Towards a Paradigm for continuous Auditin1.doc*, 2003. [cited at p. 39, 41]

[61] Andrei Ostrovski and et. al. Openl tablets - easy business rules, 2004-2014. [cited at p. 81]

[62] I. Pedrosa and C. Costa. Computer assisted audit tools and techniques in real world: Caatts applications and approaches in context. *International Journal of Computer Information Systems and Industrial Management Applications*, 4:161–168, 2012. [cited at p. 21]

[63] J. Peirson. Continuous monitoring and continuous auditing: From idea to implementation. Technical report, Deloitte & Touche LLP, 2010. [cited at p. 32]

[64] I. Rasovska, B. Chebel-Morello, and N. Zerhouni. Process of s-maintenance: decision support system for maintenance intervention. *Emerging Technologies and Factory Automation*, pages 8–15, 2005. [cited at p. 79]

[65] Z. Razaee, A. Sharbatoghlie, R. Elam, and P. Mcmickle. Continuous auditing: building automated auditing capability. *AUDITING: A Journal of Practice & Theory*, 21(1):147–163, 2002. [cited at p. 21, 29, 38, 41]

[66] C. Santos, P. Sousa, C. Ferreira, and J. Tribolet. Conceptual model for continuous organizational auditing with real time analysis and modern control theory. *Journal of Emerging Technologies in Accounting*, 5:37–63, 2008. [cited at p. 21, 38, 39]

[67] D. Searcy, J. Woodroof, and B. Behn. Continuous audit: The motivations, benefits, problems, and challenges identified by partners of a big 4 accounting firm. *36th Hawaii International Conference on System Sciences*, 1:1–10, 2003. [cited at p. 21]

[68] C. Silva and P. Carreira. Selecting audit targets using benfords law. *ISSN: 1645-2631*, 8:1–23, 2011. [cited at p. 46]

[69] L. Stanjanovic. *Methods and Tools for Ontology Evolution.* PhD thesis, Thse de doctorat de luniversit de Karlsruhne, 2004. [cited at p. 87]

[70] W. Steeman. Incident management log of volvo it belgium, 2013. [cited at p. xvii, 90, 91, 120, 160, 170, 171, 177]

[71] N. Subhani and R. D. Kent. Novel design approach to build Audit Rule Ontology for Healthcare Decision Support Systems. *Int'l Conf. e-Learning, e-Bus., EIS, and e-Gov. (EEE'14)*, pages 133–138, 2014. [cited at p. xvi, 4, 76]

[72] N. Subhani and R. D. Kent. Continuous Process Auditing (CPA): an Audit Rule Ontology approach to Audit-as-a-Service. *IEEE International Systems Conference (SysCon'15)*, pages 832–838, 2015. [cited at p. 142, 149]

[73] S. Sutton. Enterprise systems and the re-shaping of accounting systems: a call of research. *International Journal of Accounting Information Systems*, 7:1–6, 2006. [cited at p. 78]

[74] R. Valencia-Garcia, J. Ruiz-Snchez, P. Vivancos-Vicente, J. Fernndez-Breis, and R. Martinez-Bjar. An incremental approach for discovering medical knowledge from texts. *Expert Systems with Applications*, 26(3):291–299, 2004. [cited at p. 3, 59]

[75] B.F. van Dongen. Anonymized event log of a dutch academic hospital, 2011. [cited at p. xvi, xvii, 63, 64, 90, 100, 159, 162, 165, 172, 177, 178]

[76] B.F. van Dongen. Event log of a loan application process of a dutch financial institute, 2012. [cited at p. xvii, 90, 91, 93, 109, 160, 166, 169, 177, 178]

[77] M. Vasarhelyi, M. Alles, and A. Kogan. Principles of analytic monitoring for continuous assurance. *Journal of Emerging Technologies in Accounting*, 1(1):1–21, 2004. [cited at p. 21]

[78] M. Vasarhelyi, M. Alles, S. Kuenkaikaewa, and J. Littley. The acceptance and adoption of continuous auditing by internal auditors - a micro analysis. *International Journal of Accounting Information Systems*, 13:267–281, 2012. [cited at p. 21, 29]

[79] M. Vasarhelyi, M. Alles, and K. Williams. Continuous assurance for the now economy - a thought leadership paper for the institute of chartered accountants in australia. *Rutgers Business School*, pages 1–71, February, 2010. [cited at p. 21, 28, 32]

[80] M. Vasarhelyi and F. Halper. The continuous audit of online systems. *Auditing: A Journal of Practice and Theory*, 10(1):110–125, 1991. [cited at p. 20, 21, 27, 30, 78]

[81] M. Vasarhelyi, S. Romero, S. Kuenkaikaew, and J. Littley. Adopting continuous audit - continuous monitoring in internal audit. *ISACA Journal*, 3(1):1–5, 2012. [cited at p. 78]

[82] M. Vasarhelyi, R. Teeter, and J. Krahel. Audit education and the real-time economy. *Issues in Accounting Education*, pages 405–423, 2010. [cited at p. 32]

[83] L. Verner. The challenge of process discovery. *BPTrends*, May:1–11, 2004. [cited at p. 141]

[84] C. Verschoor. Continuous auditing: An operational model for internal auditors. *Internal Auditing*, 21(2):43–44, 2006. [cited at p. 21]

[85] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hubner. Ontology-based integration of information - a survey of existing approaches. *Proceedings - IJCAI Workshop*, pages 108–117, 2001. [cited at p. 4, 59, 75]

[86] Wikipedia. Wikipedia - the free encyclopedia, 2014. [cited at p. 45, 47]

[87] J. Woodroof and D. Searcy. Continuous audit - model development and implementation within a debt covenant compliance domain. *International Journal of Accounting Information Systems*, 2(2001):169–191, 2001. [cited at p. 21, 37, 40, 41]

[88] Jia Wu. *Continuous tests of details and analytical procedures in continuous auditing*. PhD thesis, Rutgers, The State University of New Jersey, 2006. [cited at p. 45]

[89] Huanzhuo. Ye, Shuai. Chen, Fang. Gao, and Yuning He. SOA-based conceptual model for continuous auditing : A discussion. *Applied Computer & Applied Computational Science (ACACOS '08)*, pages 400–405, 2008. [cited at p. 41, 42]

Vita Auctoris

NAME:	K M Numanul Hoque Subhani
PLACE OF BIRTH:	Cox'sbazar, Bangladesh
YEAR OF BIRTH:	1979
EDUCATION:	University of Windsor Windsor, Ontario, Canada 2010-2016 Ph.D. 2008-2009 M.Sc. 2007-2008 B.C.S. (Honours) 2000-2003 B.C.S. (General)

His research interests include: Continuous Auditing of Healthcare Decision Support and Transport Logistics Systems; Continuous Process Auditing; Rule and Policy-based authorization for distributed heterogeneous data sources; Supply-chain and Logistic Information System optimization; Parsing random heterogeneous datasets and Big Data Analytics, Software-as-a-Service (SaaS) application and its security in Cloud Computing Ecosystem; Strategic decision making in Healthcare Decision Support Systems; DNA microarray data analysis; and pattern recognition.