INSTITUTO
SUPERIOR
TÉCNICO

Analysis of Business Process Models based on Business-Action Theory and DEMO

Aurélio Assis

Dissertation for the degree of Master in
Information Systems and Computer Engineering

Jury Committee

President: Prof. José Carlos Martins Delgado
Adviser: Prof. Artur Miguel Pereira Alves Caetano
Evaluation Jury: Prof. Jose Manuel Nunes Salvador Tribolet

May 2011

Abstract

The existence of diversified backgrounds, terminologies, knowledge, tools, and techniques can be a significant obstacle to consensual representations of business processes within organizations. In fact, the development of such representations encompassing the correct universal patterns of communication between humans is still hard to achieve as there are very few methods and tools devoted to this purpose. The choice of business process representations may not be consensual, in terms of the modelling tools and techniques used to express them, however it can be consensual in terms of recognizing that humans always reproduce the same patterns of communication when requesting and providing services and products. Acknowledging that is the first step towards a shared platform of understanding between people interested in the representation of business processes, as the main question isn't, after all, "*how to draw the model*", but instead, "*what to draw in the model*". So this shared platform can be achieved through *Enterprise Ontology* (EO). This way, EO can assist managers, information system designers, and IT specialists to enable understanding and communication despite context and domain differences.

Keywords: Enterprise Ontology, Contract, Contract Template, Responsibility, Business Process, Circular Modelling Method.

Resumo

Frequentemente, as organizações têm de lidar com o facto de que a existência de diversos backgrounds, conhecimentos e técnicas, revela-se um obstáculo a representação consensual de modelos de processos negócio. De facto, o desenvolvimento de representações destes modelos que incorporem correctamente padrões universais de comunicação humanos ainda é algo difícil de alcançar, visto que, existem poucos métodos e ferramentas desenvolvidos para este efeito. A escolha de representações de modelos de processos de negócio pode não ser consensual no que diz respeito ao uso de ferramentas e técnicas de modelação para expressar fluxos e actividades de negócio, mas pode ser consensual quando se reconhece que os seres humanos reproduzem sempre os mesmos padrões de comportamento quando solicitam e fornecem serviços e produtos. Tal abre caminho para a introdução de uma plataforma de entendimento entre as partes interessadas na representação de um modelo, já que, a pergunta que se coloca não é "como desenhar o modelo", mas sim, "o que desenhar no modelo". Esta plataforma de entendimento pode ser alcançada por via da Ontologia Empresarial. Um modelo ontológico não dispensa outras técnicas de modelação, pelo contrário, complementa-as reduzindo a confusão terminológica e conceptual, bem como, ajudando as organizações a centrar-se no desenho de modelos que capturem devidamente os tais padrões de comportamento humanos, quando estes solicitam e fornecem produtos e serviços. Desta forma, a Ontologia Empresarial pode auxiliar gestores, projectistas de sistemas de informação e especialistas das TI na activação da compreensão e da comunicação, independentemente da diversidade de contextos ou domínios.

Palavras-chave: Ontologia Empresarial, Contracto, Modelo de Contracto, Responsabilidade, Process de Negócio, Método de Modelação Circular.

Acknowledgements

First I'd like to thank Professor Artur Caetano for advising me in this work and for giving me the opportunity to meet DEMO and this marvellous theory which is Enterprise Ontology.

I would also like to thank and dedicate this work:

to my lover and friend, Cristina Assis, for all the love and support;

to my baby daughter, "*pequena" Lueji*;

to my father, Manuel Assis, and my mother, Cecília Assis, for loving me and supporting me always;

and finally, to my lovely sister.

List of Acronyms

Acronym	Portuguese	English
	Comarca.	Judicial District.
---	Inquérito.	Inquest.
---	Polícia Municipal de Lisboa.	Lisbon Municipal Police.
AJ	Auditores Jurídicos.	Legal Auditors.
ASAE	Autoridade de Segurança Alimentar e Económica.	Economic and Food Safety Authority.
CC	Conselho Consultivo.	Justice Advisory Council.
CMVM	Comissão do Mercado de Valores Mobiliários.	Securities market Commission.
CSMP	Conselho Superior do Ministério Público.	Higher Council of the Attorney General's Office
DCIAP	Departamento Central de Investigação e Acção Penal.	Central Bureau of Investigation and Prosecution.
DGCI	Direcção Geral dos Impostos.	Directorate-General for Taxation and Finance
DGSP	Direcção Geral dos Serviços Prisionais	Directorate-General of Prison Services
DIAP	Departamentos de Investigação e Acção Penal.	Department of Investigation and Prosecution.
DJ	Distrito Judicial.	Judicial district.
GDDC	Gabinete de Documentação e Direito Comparado.	Bureau of Documentation and Comparative Law.
GNR	Guarda Nacional Republicana.	Republican National Guard.
INML	Instituto Nacional de Medicina Legal	National Institute of Forensic Medicine
IRN	Instituto dos Registos e do Notariado	Institute of registries and notaries
MP	Ministério Público	Prosecutor
NAT	Núcleo de Assistência Técnica.	Technical Assistance Bureau.
OPC	Órgãos de Polícia Criminal.	Criminal Police Agencies.
PGD	Procuradoria Geral Distrital.	District Attorney.
PGR	Procuradoria Geral da República.	Attorney General’s Office

PJ	Polícia Judiciária.	Judicial Police.
PJM	Polícia Judiciária Militar	Military Judicial Police
PSP	Polícia de Segurança Pública.	Public Security Police.
SEF	Serviço de Estrangeiros e Fronteiras.	Immigration Services.
STA	Supremo Tribunal Administrativo.	Supreme Administrative Court (**SAC**).
STJ	Supremo tribunal de Justiça.	Supreme Court.
TC	Tribunal de Contas.	National Audit Office.
TCA	Tribunal Central Administrativo.	Central Administrative Court.
TCnst	Tribunal Constitucional.	Constitutional Court.
TR	Tribunal da Relação.	Court of Appeal.

Addendum

The following chapters were revised as required by the jury committee:

Chapter 2 (Thesis Problem): New topics were added (*section 2.1.1 to section 2.1.2*). The problems to be addressed were described in a more comprehensive manner.

Chapter 3 (Thesis Statement): New topics were added (*section 3.2 to section 3.3*). The concepts of *contract* and *responsibility* were introduced and discussed.

Chapter 6 (Circular Method For Analysis and Compliance): The semantic analysis of business process models (e.g. form BPMN, Activity Diagrams, etc.) was revised and theoretically reasoned.

Chapter 7 (Case Study – The Attorney General's Office): The gap analysis between the BPM and the resulting DEMO model was revised.

Chapter 8 (Conclusions): Future work discussion (Section 8.1.3) was revised.

Table of content

List of Figures

List of Tables

1

Introduction

The interesting question regarding systems based approaches to organizational design is: what kinds of questions would be normally left out from any specific approach? Practical reasoning is always open textured.

Garrett Thomson (Philosopher)

With the arrival of Business Process Reengineering (BPR), Total Quality Management (TQM) and ISO 9000, the documentation and adoption of models to design and represent business processes has proven to be a constantly unfinished and continuous work. It means that organizations often need to improve the way they document and model their business processes. The reality shows that modelling tools seem not to have all the answers when it comes to analyse and improve business process models.

Business process models are often represented by standard modelling methods such as UML modelling language (activity diagrams) (Rumbaugh, Jacobson, & Booch, 1999), BPMN (Group, 1997-2011) and sometimes (in many practitioner cases) by ad-hoc diagrams with some notion of activity. These graph-oriented modelling methods allow us to represent the sequence of activities and sometimes the actors performing them. In fact, the knowledge provided by business process models is extremely relevant as it is the best way to show how a particular business case should be carried out (Gordijn, Akkermans, & Vliet, 2000).

This thesis work provides a method that relies on Enterprise Ontology and DEMO to enhance graph-oriented business process models. The goal is to verify their completeness[1] and correctness[2] trough a PIF/CAP-analysis[3] over acts of speech.

[1] Every acts of speech are defined and described according to universal patterns of communication as stated by the Ψ-theory (Dietz, 2006)

1.1 Motivation

In the last decade business process models are receiving more attention as the interest for process centric representations of organizations is becoming more widespread. Whereas earlier mostly data-driven approaches have been pursued as a starting point for information systems modelling, there is currently a tendency to use more process-driven requirements engineering (Nuffel, Mulder, & Kervel, 2009). This trend has even increased by recent developments like Service-Oriented Architectures (SOA) where business process languages are considered as primary requirements sources (Zimmermann, Schlimm, Waller, & Pestel, 2005).

To model business processes a large number of notations, languages and tools exist as a result of the increasing use of business process models to analyse business processes in the early phases of systems development (Dijkmana, Dumas, & Ouyang, 2008). The research made in this work was motivated by the need to produce unambiguous and coherent models of business processes, assuring its completeness and correctness. This need is due the fact that graph-oriented modelling representations of these models often have some drawbacks. Regarding the basic assumptions of current business process modelling languages, following remarks are mentioned: absence of formal semantics, limited potential for verification, message-oriented approach, and the modelling of multi-party collaborations (Barjis, 2007). When analyzing Flowchart and activity based modelling languages like BPMN, for example, the lack of formal semantics is caused by the heterogeneity of its constructs, and the absence of an unambiguous definition of the notation (Dijkmana, Dumas, & Ouyang, 2008).

The evaluation of flowchart and activity based modelling representations mainly concentrates on the following relevant aspects of research: the facts concerning completeness, consistency, construct redundancy[4] and ambiguity of the modelled business process. Thus, it can be mentioned that the theoretical studies still indicate a number of problems that practitioners actually suffer from when dealing with conventional graph-oriented modelling representations (Recker, Indulska, Rosemann, & Green, 2006). Most problems are moreover indicated by people with an IT background who need more rigor and details to use models like BPMN, for example, as input for software implementation projects

---

[2] No axioms of the Ψ-theory are violated which means the corresponding ontological model is correct in terms of the operations and transactions described, and in terms of its composition.

[3] PIF stands for *Performa-Informa-Forma*, and CAP stands for *Coordination-Actors Roles-Production*. These concepts can be understood by reading (Dietz, 2006). The PIF/CAP analysis is also described in (Shishkov, Xie, Liu, & Dietz, 2005).

[4] For example, in BPMN modelling language a thing can be represented by both a pool and a lane, a transformation is represented by an activity, a task, a collapsed sub-process, an expanded sub-process, and a transaction.

(Nuffel, Mulder, & Kervel, 2009). Considering these issues, we recognise the increasing need to design correct and structurally complete business processes.

1.2 Contributions

This thesis work is expected to offer the following contributions:

1. Provide a method to demonstrate the suitability of DEMO on improving graph-oriented business process model representations.
2. Help to identify information systems requirements through documented[5] human acts of speech within the organization.
3. Facilitate discussions about redesign, as well as the improvement of business process models.

1.3 Thesis structure

This thesis work includes eight chapters as illustrated by Fig. 1.1.

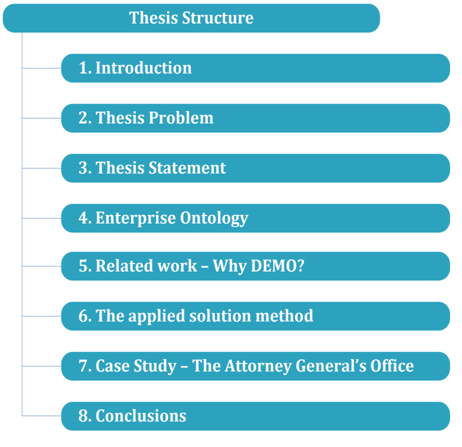

Figure 1.1 – Thesis outline.

Chapter 2, *Thesis Problem,* introduces the thesis application scenario and describes the problems that we are aiming to address.

Chapter 3, *Thesis Statement,* introduces the research questions, the adopted research method, and the thesis goal.

Chapter 4, *Enterprise Ontology*, describes the theory used to support the research work and validate the thesis case study.

Chapter 5, *Related work – Why DEMO,* presents a brief discussion about business process modelling languages and makes some comparisons between DEMO and BPMN and then between DEMO and Archimate.

[5] Developed and designed business process models.

Chapter 6, *The applied solution method,* presents the C*ircular Method for Analysis and Compliance* (CMAC) we developed to improve graph-oriented business process models. This method is based on DEMO methodology which is described in chapter 4

Chapter 7, *Case Study – The Attorney General's Office*, describes the thesis case study and demonstrates the application of the CMAC in the context of the Inquest Opening Process in the Portuguese Investigation Department (DIAP).

Finally, chapter 8 presents the research conclusions and a final discussion of the results achieved.

2

Thesis Problem

According to Hevner, (2004), the objective of research in information systems is to acquire knowledge and understanding that enables the development and implementation of technology-based solutions to heretofore unsolved and important business problems. Hevner, (2004) then claims that design science approaches this goal through the construction of innovative artefacts aimed at changing the phenomena that occur.

By following the design science approach, this research effort is with respect to practitioners who plan, manage, design, implement, operate, and evaluate information systems. It addresses a set of problems faced by that community when using flow-chart based modelling languages, such as BPMN, to describe graphically business processes that will be supported by information systems.

In terms of research, the implications of the problems identified in this chapter justify the developing of an effective artefact that enables such problems to be addressed – This artefact is the method by which those problems will be analysed and approached. So the following section describes each of these raised issues, revealing some of the most relevant drawbacks and limitations of the current modelling languages.

2.1 Limitations of Current Modelling Languages

Graph-oriented BP modelling languages such as BPMN, for example, have some drawbacks regarding aspects like the way models are produced. One of the most relevant is the absence of rigid methods to define its constructs as well as the lack of a formal and explicit specification of a shared conceptualization. Other aspects have to do with the lack of a complete choreography these models may suffer from when the representation of the involved business processes do not reflect explicitly the correct *transactional patterns*[6] of human acts of communication. Other three serious aspects that we will address have to do with lack of formal semantics, and incompatible, as well as, redundant representations of business processes in the model.

[6] These universal patterns regard to three phases in which a transactions evolves: A *Request Phase*, followed by an *Execution Phase*, which is then followed by a *Result Phase* (Dietz, 2006).

2.1.1 Lack of Rigid Modelling Methodologies

Most of the adopted graph-oriented business process modelling languages lack from a rigid methodology as is the case of BPMN which intends to be methodology agnostic. This characteristic brings some important advantages such as flexibility but at the same time leaves room for some inconvenient. The most evident is when the modelling of an enterprise becomes a tough job as the available descriptions of its operations tend to be voluminous and at the same time incomprehensive and inconsistent. Therefore, it may be imperative to develop graph-oriented business process models under a rigid design methodology that allows us to overcome problems caused by confusion and complexity of these models. As we will see, this methodology is DEMO[7].

2.1.2 Lack of Formal Semantics

Graph-oriented process definition languages or simply flowchart-based modelling languages, have been previously studied from a formal perspective in (Aalst & Arthur, 2000). It is known that these families of modelling languages, which includes BPMN, can exhibit a range of semantic errors, including deadlocks and livelocks[8] due the inexistence of a defined formal semantic of their notations (Dijkmana, Dumas, & Ouyang, 2008). The BPMN, for example, the most popular graph-oriented process definition language, is a standard largely used for capturing business processes in the early phases of systems development. Specialists often realise that the mix of constructs found in BPMN makes it possible to obtain models with a range of semantic errors (Dijkmana, Dumas, & Ouyang, 2008). Thus, the ability to statically check the semantic correctness of models is a desirable feature for modelling techniques and tools based on graph-oriented business process modelling languages like BPMN. However, the static analysis of these modelling languages is hindered by its ambiguities and potential complexity. In the case of BPMN, for example, the fact that it integrates constructs from modelling languages with features for concurrent execution of multiple instances of a sub process and exception handling makes it challenging to provide a formal semantics (Dijkmana, Dumas, & Ouyang, 2008).

While syntactic rules can be comprehensively documented, the actual semantics of these modelling languages is only described in narrative using sometimes inconsistent terminology. This work takes on the challenge of defining a method that overcomes these issues by using semantic analysis as proposed in (Xie, Liu, & Dietz, 2005) which is soundly rooted in the semiotic theory developed for information systems in (Liu, 2000).

2.1.3 Lack of complete choreography

Process choreography describes the interactions between different business actors and the dependencies between these interactions. It indicates the actions or steps each involved party

[7] DEMO is an acronym for Design and Engineering Methodology for Organizations.
[8] A specific process is not progressing.

performs in the interaction. While there can be different proposals for capturing choreographies within graph-oriented business process models, it remains unclear how these choreographies should be described on a conceptual level. In the case of BPMN, for example, choreography can be used by expressing interconnected interface behaviour models (Decker & Barros, 2008). However, this modelling style often leads to redundant dependencies and the danger of incompatible processes (Decker & Barros, 2008).

Within a business process, choreography involves the actions of two or more actor roles aimed at achieving a particular result. The problem we present in this section is how these actions are related to each other. Are there particular structures or patterns in which they occur, is there any pattern possible? If there are any structures of patterns how can we make sure that interactions expressed within a business process model reflect correctly such patterns? Actually no graph-oriented process definition language assures that property.

2.1.4 Incompatibility of Processes

One of the most important drawbacks of graph-oriented business process models can be the lack of compatibility of the processes represented in the model when analysing the complete composition environment. To avoid this problem, modellers often need to perform an accurate analysis process to assure that business process interactions will be compatible and therefore match the requirements of interaction. An example for process incompatibility would be a supplier who waits for the payment to arrive before delivering the purchased goods. The buyer, on the other hand, waits for the goods to be delivered before actually paying for them. Both partners would wait endlessly – a classical bad design situation. This means that the dependency between both processes is not explicitly described in the model.

This problem can be overcome if we take advantage of the rigor and accuracy of the DEMO methodology to develop complete and correct business process models.

2.1.5 Unclear Contracts and Responsibilities

Business process models are typically used to express inter or intra-enterprise business activities/processes (Kabilan, 2005). Contractual obligations need to be fulfilled through execution of business processes on behalf of the contracting parties (Kabilan, 2005). To do so, business contract terms and conditions need to be semantically integrated to existing business process models (Kabilan, 2005). Contract obligations, contract results, contract deadlines and other related concepts are often hard to make explicit within graph-oriented business process models like BPMN for example. To face this problem, we propose the use of DEMO for reducing a business process model to its corresponding DEMO transactions[9] giving us a transactional and compositional structure of the analysed business process. Through DEMO we can see a business process from a different perspective, as nothing but a set of

[9] For example, to identify every DEMO transactions from a previously designed BPMN model.

contracts between business subjects[10] and each contract is fulfilled by a single business transaction[11]. We believe that this new way of thinking is the key to enable information systems designers to focus on different transactions of a business process and even develop different parts of the system to support those transactions in a concurrent but controlled way. It then requires the formalisation of a contract specifying its different aspects such as the partakers of the contract, their responsibilities, the results to be achieved as the way to achieve them, and a uniform representation of the information objects used in the contract and required by or passed to other contracts. In chapter 6 we will describe the concepts involved in the formalisation of a contract and how it can be used to represent in a structured and organized way human interactions within a business process; a goal that is hard to achieve by simply looking at business process models such as BPMN. So the formalization of contracts at the business level aims to create a short path from business process models design and analysis, to information systems requirements analysis and development.

2.2 The Application Scenario – Thesis Validation

To illustrate the thesis problem and to validate the subsequent design artefact, we propose the reader to look at chapter 7 which describes the thesis case study. The case study is focused on one of the business process models of the Portuguese Investigation Department (DIAP), namely, the *Inquest Opening process*, which describes the submitting of a complaint from a citizen or other external entity.

During the development of the information system[12] (IS) responsible for supporting the management and life cycle of the inquest in the Portuguese Investigation Department (DIAP), a set of models were produced as the input for requirements analysis of that system. Our mission in this work is to validate a generic method based on DEMO by demonstrating that it can be applied to correct/improve graph-oriented business process models, and to formalise contracts which can be used to produce IS requirements and enable its concurrent and controlled development by system designers and developers.

[10] Human beings.
[11] A DEMO transaction.
[12] The SIMP-NG. For more information please visit http://www.itij.mj.pt/PT/Paginas/Default.aspx

3

Thesis Statement

In order to formulate the thesis hypotheses we find imperative to discuss some of the core concepts that will be of great use for our research purposes. So let's start by defining business process model.

3.1 Business Process Model

Business Process Modelling activities by implication focus on processes, actions/activities, resources and people (e.g. teams or departments). Thus, a Business Process Model (BPM) is typically developed by business analysts and managers who are seeking to improve process efficiency and quality. The process of design and improvement of BPMs may or may involve the development of information systems, although that is a common driver for the need to model a business process. The Business Process Modelling Notation (BPMN), for example, is one the most used modelling language to model organizations, and a good reference for the type of BPMs that will be the object of our research.

In this work, we will refer to Business Process Models (BPM) as being graph-oriented representations of sequences of *activities*, typically showing *events*, *actions* and *links* or *connection points* in those sequences from end to end. So our research work targets BPM representations based on generic design structural elements typically used in flowchart diagrams:

- Decision points (*gateways*).
- Actors (pools and swimlanes).
- Activities (The name of the activity).

In the next chapters we will cover with more detail other important aspects of business process models and modelling languages.

3.2 Contract

A contract is often defined as an enforceable agreement between two or more parties with mutual obligations. The term is commonly used in social disciplines to describe (legal) bonds between two or more social entities. In the last years the term contract is becoming more popular in several industrial sectors including the IS industry.

In the e-commerce industry web services technology is a practical example of how the notion of contract is applied to enable technological solutions in a standardized manner. A web service is defined through a contract known as WSDL[13] which is an XML format describing the available network services or transactions (Christensen, Erik; Curbera, Francisco; Meredith, Greg; Weerawarana, Sanjiva, 2001).

About Enterprise Ontology (EO) the notion of transaction is at the centre of business processes within organizations. However, the notion of contract is left implicit. At the business process level a contract always implies the execution of a transaction. From EO theory we know that a business process is fulfilled through one or more transactions[14] carried out by social individuals (humans). From this assumption, we state that if a business process is fulfilled through one or more ontological transactions, then each transaction in that business process can be part of a particular contract. As we demonstrate in *chapter 6* a contract will describe a number of essential organizational aspects, such as deadlines, transactions, locations, people's responsibilities, and goals. This definition of contract is essentially based on human interactions within social systems and attempts to describe and formalise those interactions on the basis of enterprise ontology theory and DEMO. This way a contract can be a valuable artefact for *information systems* analysts by enabling independent discussions over different fragments of a business process including the actions people perform in it as they are the potential users of any future delivered *information system*.

3.3 Responsibility

As stated by Dietz, (2006) we also strongly oppose the quite common idea that artificial agents are, or at some future time will be, equivalent fellow players in human social systems. This idea can only be justified by a severe inflation of such notions as authority and responsibility. Within social systems responsibility becomes manifested primarily in coordination between subjects (humans) who are the elementary actor roles or atomic amounts of authority and responsibility.

In this work we outline the importance of addressing DEMO's responsibility, competence and authority notions at the design level in order to enable the process of analysis and improvement of BPMs. This can be particularly important in areas like Enterprise Governance (EG) as shown by (Henrique, 2010).

Therefore, we intend to demonstrate the importance of redesigning BPMs by defining ontological models and by using the underlying notion of responsibility.

[13] Web Services Description Language.

[14] Please see the ontological definition of business process and transaction in chapter 4 section. 4.1.2.

3.4 Thesis Hypothesis

We now formulate the thesis hypothesis:

> ***Enterprise ontological models defined with DEMO provide essential concepts that can be used to improve business process models, as well as, to support and formalise the notion of business process contract.***

3.5 Research Method

The thesis research method that was used is Design Science Research (DSR) for IS. The fundamental principle of design-science research from which seven guidelines are derived is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artefact (Hevner, 2004). That is, design-science research requires the creation of an innovative, purposeful artefact (Guideline 1) for a specified problem domain (Guideline 2). Because the artefact is *purposeful*, it must yield utility for the specified problem (Hevner, 2004). Hence, thorough evaluation of the artefact is crucial (Guideline 3). Novelty is similarly crucial since the artefact must be *innovative*, solving a heretofore unsolved problem or solving a known problem in a more effective or efficient manner (Guideline 4) (Hevner, 2004). In this way, design-science research is differentiated from the practice of design. The artefact itself must be rigorously defined, formally represented, coherent, and internally consistent (Guideline 5) (Hevner, 2004).

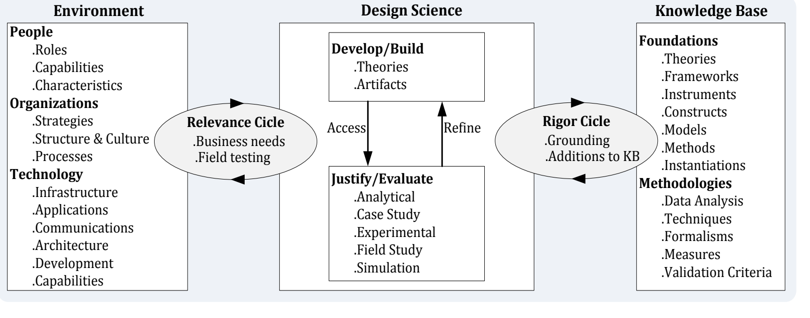

Figure3.1 – Design Science Research Methodology.

The process by which it is created, and often the artefact itself, incorporates or enables a search process whereby a problem space is constructed and a mechanism posed or enacted to find an effective solution (Guideline 6) (Hevner, 2004). Finally, the results of the design-science research must be communicated effectively (Guideline 7) both to a technical audience (researchers who will extend them and practitioners who will implement them) and to a managerial audience (researchers who will study them in context and practitioners who will decide if they should be implemented within their organizations) (Hevner, 2004). Fig. 3.1 Illustrates the DSR methodology.

Table 3.1 – Design-Science Research Guidelines

Guideline	*Description*
1: Design as an Artefact	*Design-science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation.*
2: Problem Relevance	*The objective of design-science research is to develop technology-based solutions to important and relevant business problems.*
3: Design Evaluation	*The utility, quality, and efficacy of a design artefact must be rigorously demonstrated via well-executed evaluation methods.*
4: Research Contributions	*Effective design-science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies.*
5: Research Rigor	*Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artefact.*
6: Design as a Search Process	*The search for an effective artefact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.*
7: Communication of Research	*Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.*

3.6 Research Artefacts

According to the first guideline of (Hevner, 2004), a design-science research must produce a viable artefact in the form of a construct (vocabulary and symbols), a model (abstractions and representations) a method (algorithms and practices), or an investigation (implemented and prototype systems).

In order to support and validate this thesis the following artefacts were clearly conceived:

1. A **set of DEMO models** produced from the input business process model (a flowchart based representation) of the DIAP. The following DEMO models were produced[15]:
   - *i. Process Model (PM) – Includes the Process Structure Diagram (PSD) and the Information Use Table (IUT).*
   - *ii. State Model (SM) – Includes the Object Fact Diagram (OFD) and the Object Property List.*

[15] For more information about DEMO models and Enterprise Ontology we recommend to take a look at chapter 4 or to consult (Dietz, 2006, pp. 139-213).

   iii. *Construction Model (CM) – Includes the Action Transaction Diagram (ATD) and the Transaction Result Table (TRT)*
2. The thesis **applied solution method**, named Circular Method for Analysis and Compliance (CMAC). The CMAC uses the DEMO methodology to produce an ontological model out of flowchart and activity based business process model representations. Then, it uses the resulting ontological model (DEMO models) to refine and improve the input flowchart and activity based business process models.
3. The **validation scenario**, where the thesis solution method indicated above is applied in the context the Inquest management carried out by the Investigation Department (DIAP), which is under the authority of the Portuguese Attorney General's Office.
4. A **Refined and improved business process model of the DIAP** compliant with the Ψ theory (Dietz, 2006) and produced from applying the CMAC. This model is a result of the validation scenario.
5. The formalization of a contract through a **contract template,** which is a document that indicates every aspects of a business process transaction between two entities.

3.7 Research Questions

This section presents the four research questions we intend to address in this thesis work:

1. *Can we use DEMO to identify inconsistencies and improve graph-oriented business process models[16]?*
2. *Is DEMO suitable to formalise contracts describing human interactions within a business process?*

[16] Such as BPMN models, for example.

4

Enterprise Ontology

This chapter introduces the core concepts necessary for conducting the thesis project. All concepts described in this chapter are based on Jan Dietz work, *Enterprise Ontology* (Dietz, 2006). The purpose of this chapter is to set a relevant understanding about the concept of *enterprise ontology*, and introduce the *ontological model* (DEMO). Enterprise ontology gives us an interesting perspective about organizations. It allows us to better understand the construction of organizations at the level of human interactions. The ontological model gives us a methodology for modelling and describing those interactions.
This chapter starts by introducing the Ψ (PSI) theory which is the theory behind the notion of Enterprise Ontology. Understanding the Ψ-theory is the first step to capture the essence of DEMO models. It presents us the most basic principles of human interactions within organizations, which are not very well captured by models like BPMN, and UML, for example.

4.1 The Ψ (PSI) Theory

The Ψ-theory is the theory that underlines the notion of enterprise ontology (Dietz, 2006, p. 13). An enterprise will be defined as a (heterogeneous) system in the category of social systems. Being a social system means that the *elements* are social individuals, i.e. subjects. The Ψ-theory consists of four axioms. In the following sections we describe each of these axioms.

4.1.1 The Operation Axiom

The first axiom of the Ψ-theory states that the operation of an enterprise is constituted by the activities of actor roles, which are elementary chunks of authority and responsibility, fulfilled by subjects (Dietz, 2006). In doing so, these subjects perform two kinds of acts: *production acts* and *coordination acts*. These acts have definite results: *production facts* and *coordination facts*[17], respectively (Dietz, 2006). The axiom is commonly referred to as the *Operation Axiom*.
By performing *production acts* (P-acts for short) the subjects contribute to bringing about the goods and/or services that are delivered to the environment of the enterprise (Dietz, 2006). The realization of a production act is inherently either material or immaterial. Examples of material acts are all manufacturing acts, as well as storage and transportation acts. Examples

[17] See (Dietz, 2006, pp. 35-44) for a better understanding of the notion of an ontological fact.

of immaterial acts are the judgment by a court to sentence someone, the decision to grant an insurance claim, and the appointment of someone as president (Dietz, 2006). By performing *coordination acts* (C-acts for short) subjects enter into and comply with commitments towards each other regarding the performance of production acts (Dietz, 2006). A subject in its fulfilment of an actor role is called an *actor*.

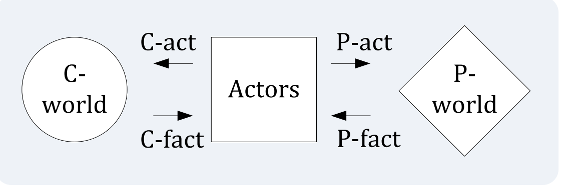

Figure 4.1 – Graphical representation of the operation axiom.

Fig. 4.1 exhibits the operation axiom graphically. The symbol for coordination is the disc, the symbol for actor roles is the box, and the symbol for production is the diamond (Dietz, 2006). The plain arrow from the actor's box to the C-world disc expresses that actors perform C-acts. The dashed arrow from the C-world disc to the actor's box expresses that actors take account of the state of the C-world when being active (Dietz, 2006). Likewise, the plain arrow from the actors box to the P-world diamond expresses that actors perform P-acts, and the dashed arrow from the P-world diamond to the actors box expresses that actors take account of the state of the P-world when active (Dietz, 2006).

4.1.2 The Transaction Axiom

A transaction actually consists of two conversations: an actagenic or order conversation, and a factagenic or result conversation (Dietz, 2006). A conversation is defined as a sequence of coordination acts between two actor roles that are aimed at achieving a well-defined result concerning a P-act/fact (Dietz, 2006). Between them is the execution of the production act, which both conversations are about. So, a transaction evolves in three phases: *the order phase (O-phase for short), the execution phase (E-phase for short), and the result phase (R-phase for short)* (Dietz, 2006). One of the two partaking actor roles is called the initiator, and the other is called the executor of the transaction. In the order phase, the initiator and the executor work to reach an agreement about the intended result of the transaction, i.e., the production fact that the executor is going to create as well as the intended time of creation (Dietz, 2006). In the execution phase, this production fact is actually brought about by the executor. In the result phase, the initiator and the executor work to reach an agreement about the production fact that is actually produced, as well as the actual time of creation (both of which may differ from what was originally requested) (Dietz, 2006). Only if this agreement is reached will the production fact come into existence. The moment at which it starts to exist is agreed upon in the result phase by the two actor roles (to be precise, between the two subjects that play the role of initiator and executor, respectively) (Dietz, 2006).

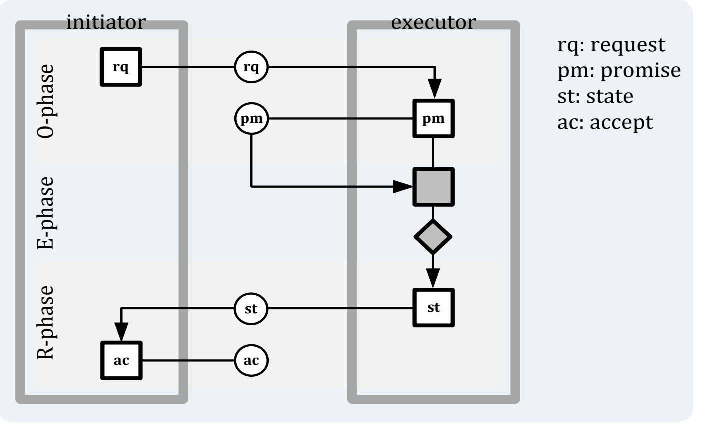

Figure 4.2 - The basic pattern of a transaction

4.1.3 The Composition Axiom

The composition axiom provides the basis for a well-founded definition of the notion of business process (Dietz, 2006), which states that a business process is a collection of causally related transaction types, such that the starting step is either a request performed by an actor role in the environment (external activation) or a request by an internal actor role to itself (self-activation). Every transaction type is represented by the complete transaction pattern (Dietz, 2006).

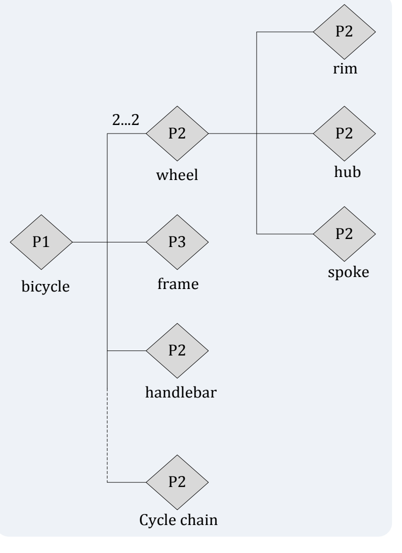

Figure 4.3 – Component structure of a bicycle.

From the previous section we have seen that the result of a successful transaction is the creation of a P-fact. Conversely, every original new fact in the P-world is brought about as the result of a successful transaction. The question addressed in this section is how these facts are interrelated. Let us, as an example, look at the manufacturing of a bicycle. As everyone knows, manufacturing a bicycle is not one (atomic) act, like switching on your desk lamp.

Instead, it is an assembly of a number of parts. Next, it is useful to distinguish between atomic parts and subassemblies. An atomic part cannot be disassembled; examples are a bolt and a nut. Most parts of a bicycle, like the lamps and the chains, are subassemblies. So, a bicycle can conveniently be conceived as a tree structure of parts, both atomic parts and subassemblies, as exhibited in Fig. 4.3.

As stated before, the creation of every part is the outcome of a successful transaction, in the case of the bicycle: either a manufacturing transaction or an acquiring (e.g., a purchase) transaction. Clearly, the logical sequence of the component structure of the bicycle, as shown in Fig. 4.3, puts restrictions on the sequence in which all these transactions are carried through. For example, to complete the production of the bicycle assembly (P1), the parts P2, P3, P4, etc., have to be produced first. Similarly, in order to complete the production of a P2, the production of a P5, a P6 and 40-60 P7s has to be completed first. The general question then is how the transaction in which an assembly is produced is related to each of the transactions in which a component is produced or acquired. The answer is shown in Fig. 4.4. It is derived from the basic pattern in Fig. 4.2.

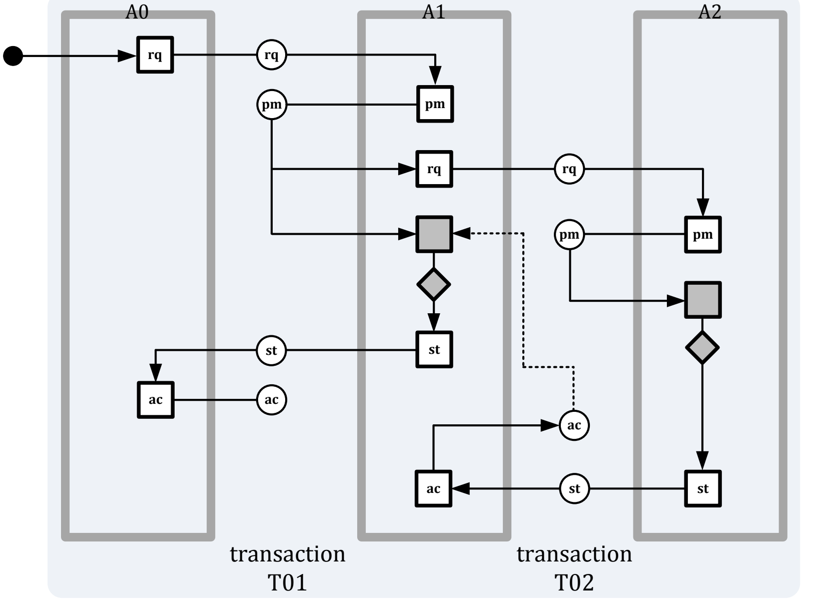

Figure 4.4 – The structure of enclosing a transaction.

4.1.4 The Distinction Axiom

The fourth axiom of the Ψ-theory states that there are three distinct human abilities playing a role in the operation of actors, called performa, informa, and forma. These abilities regard communicating, creating things, reasoning, and information processing. Because this axiom serves, in particular, in neatly separating our diverse concerns, it is called the distinction axiom. Fig. 4.5 summarizes the distinction axiom.

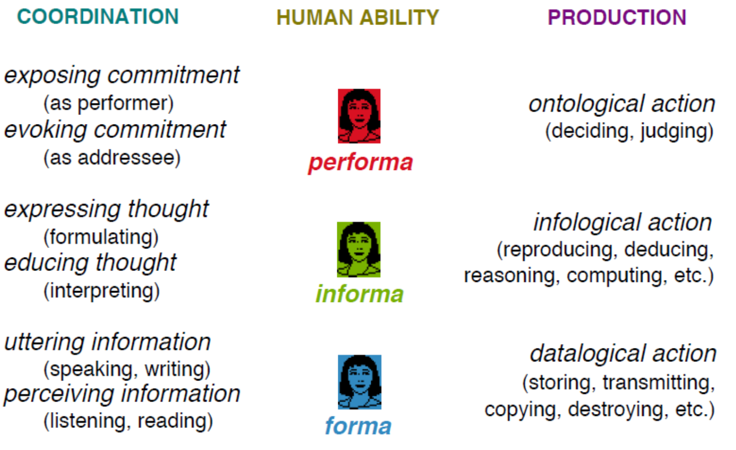

Figure 4.5 – Summary of the distinction axiom.

The *forma* ability (to perform *datalogical* acts) concerns the form aspects of communication and information. In other words, we are talking now about such things as the uttering and perceiving of sentences in some language, the syntactical analysis of such sentences, coding schemes, transmission of data, and storage and retrieval of data or documents (Dietz, 2006). For the sake of convenience, we consider the forma ability to also comprise the physical substrate in which information is encoded (Dietz, 2006). The *informa* ability (to perform *infological* acts) concerns the content aspects of communication and information ("in-forma" means, in Latin, what is in the form). So, we are dealing now with communication and information while fully abstracting from the form aspects. In other words, we are concerned with such things as the sharing of thoughts between people, the remembering and recalling of knowledge, and reasoning. The *performa* ability (to perform *ontological acts*) ("per-forma" means, in Latin, through the form) concerns the bringing about of new, original things, directly or indirectly by communication (Dietz, 2006). We are talking now about engaging into commitments, and about decisions, judgments, etc. We consider the performa ability as the *essential* human ability for doing business of any kind (Dietz, 2006).

4.2 The DEMO Methodology

This section describes the four aspect models of the DEMO methodology. Fig. 4.6 illustrates the DEMO ontological models and its hierarchy.

4.2.1 The Construction Model

The construction model (CM) of an organization specifies its composition, its environment, and its structure, according to the system definition that is provided (Dietz, 2006, pp. 57-62). The composition and the environment are both a set of actor roles. By convention, we will always draw environmental actor roles as composite actor roles, even if we happen to know that an actor role is elementary. The reason for doing this is that generally we do not know whether an environmental actor role is elementary or composite. Moreover, we do not care: our interest is in the kernel of the organization.

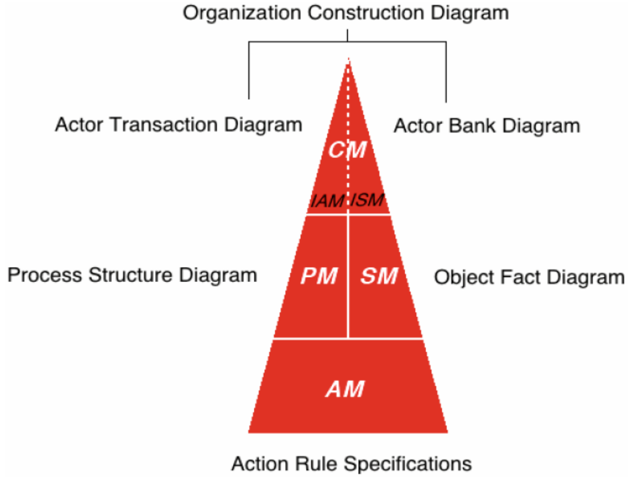

Figure 4.6 – The ontological triangle of the diagrams.

We start with modelling this kernel as one composite actor role. The resulting CM is usually referred to as the global CM of an organization. Likewise, the CM in which the kernel contains only elementary actor roles is called the detailed CM. The boundary divides the set of all (relevant) actor roles

into the composition and the environment.

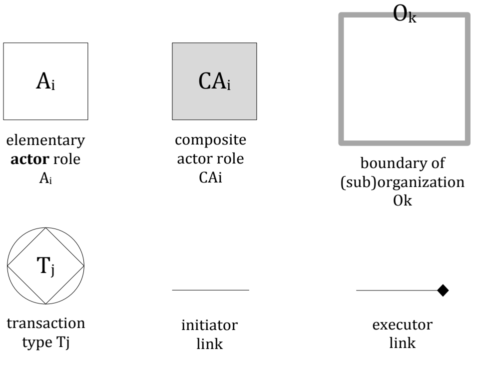

Figure 4.7 - Legend of the Actor Transaction Diagram (interaction).

The *interaction structure* of an organization consists of the transaction types in which the identified actor roles participate as initiator or executor. It is expressed in an Actor Transaction Diagram (ATD) and a Transaction Result Table (TRT). The legend of the Actor Transaction Diagram is shown in Fig. 4.7.

4.2.2 Process Model

The process model (PM) of an organization is the specification of the state space and the transition space of the C-world; thus, the set of lawful or possible or allowed sequences of states in the C-world. This is to a large extent determined by the transaction pattern. Since every transition in the C-world consists of the creation of a C-result and since there is a one-to-one relationship between this C-result and the causing C-act, these C-acts are also contained in the PM. A C-result and its causing C-act are collectively called a process step. The PM specifies also for every process step the information used to perform the step. As a convenient addition, the PM duplicates the knowledge from the CM concerning which actor roles perform the C-acts. They are called responsibility areas (see Fig. 4.8).

A process model is expressed in a Process Structure Diagram (PSD) and an Information Use Table (IUT). Although the latter cannot in fact be produced before the SM is finished, we show it early on in this chapter. One has to keep in mind, however, that in practice this is done later. Fig. 4.8 exhibits the legend of the PSD. The PSD has a more concise notation: the symbol of the C-fact type (a small disk) is pushed, so to speak, onto the symbol of the C-act type (a small box), from which it is created. This is possible since any C-act type has exactly one C-fact type as its result. A similar reasoning holds when pushing the small diamond of the P-fact type onto the small box of the P-act type. The combined symbols represent process steps. In principle, the PSD specifies for every included transaction type the process steps that are allowed to be taken. Consequently, steps that are not included in the PSD are not allowed.

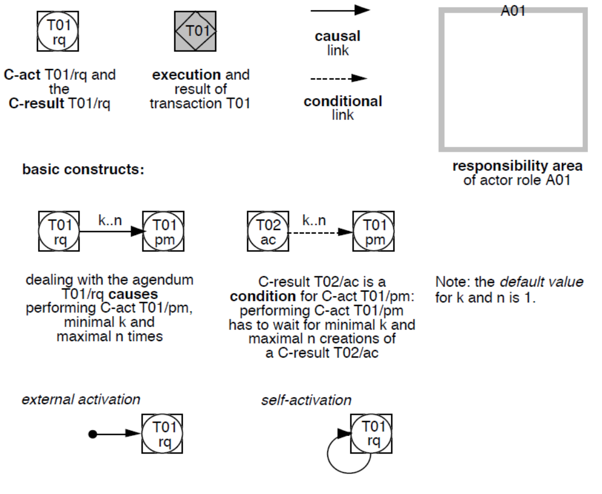

Figure 4.8 – Legend of the Process Structure Diagram.

The PSD of a business process should be understood as the complete specification of the steps in a business process that an enterprise wants to monitor or control. As such, the PSD of a business process is the right starting point for the design of workflow support systems.

4.2.3 Action Model

The action model (AM) is the most detailed and comprehensive aspect model. It is also atomic on the ontological level. Strictly speaking, the other three aspect models (PM, SM, and CM) are derived from the AM. One should keep in mind, however, that action rules serve only as guidelines for an actor. Sometimes it might be necessary for an actor to deviate from an action rule. Ultimately, the actor is held responsible for his or her acting.

4.2.4 State Model

The state model (SM) of an organization is the specification of the state space of the P-world. It consists of specifying the object classes, the fact types, and the result types, as well as the existential laws that hold. An SM is expressed in an Object Fact Diagram (OFD) and an Object Property List (OPL). The OPL is just a convenient way of specifying fact types that are proper (mathematical) functions, and of which the range is a set of values. One may as well specify them in an OFD, but that would make the OFD unnecessarily voluminous. The fact types in an OPL are called properties (of object classes). Its legend is presented in Fig. 4.9 and 4.10.

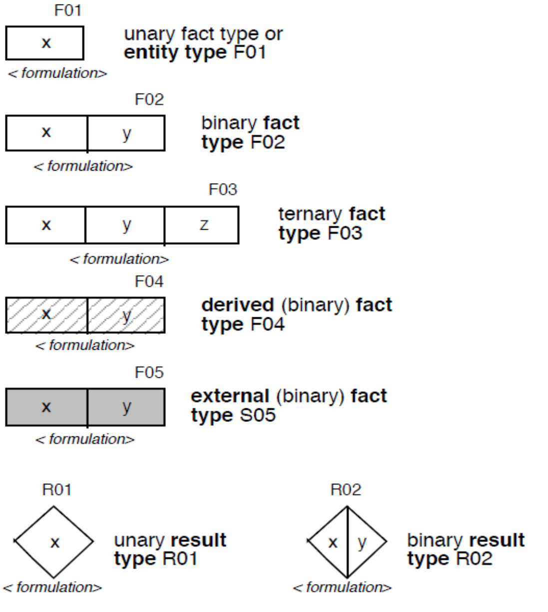

Figure 4.9 - Legend of the Object Fact Diagram (first part).

In principle, one can find the categories, object classes, fact types, and result types, as well as all pertaining existential laws and all derivation rules, in the action rule specifications (Dietz, 2006). The contents of both the OFD and the OPL of an organization are completely determined by its action model. This makes the state model of an organization a truly objective model: only the information items that are relevant for the operation of the organization are included (Dietz, 2006). This is in sharp contrast to the current practice in requirements engineering, in which the information wishes of users are collected. This way of working, which we like to refer to as the waiter strategy, leads on the one hand to incompleteness (absence of necessary information) and on the other hand to over completeness (presence of unnecessary information) (Dietz, 2006).

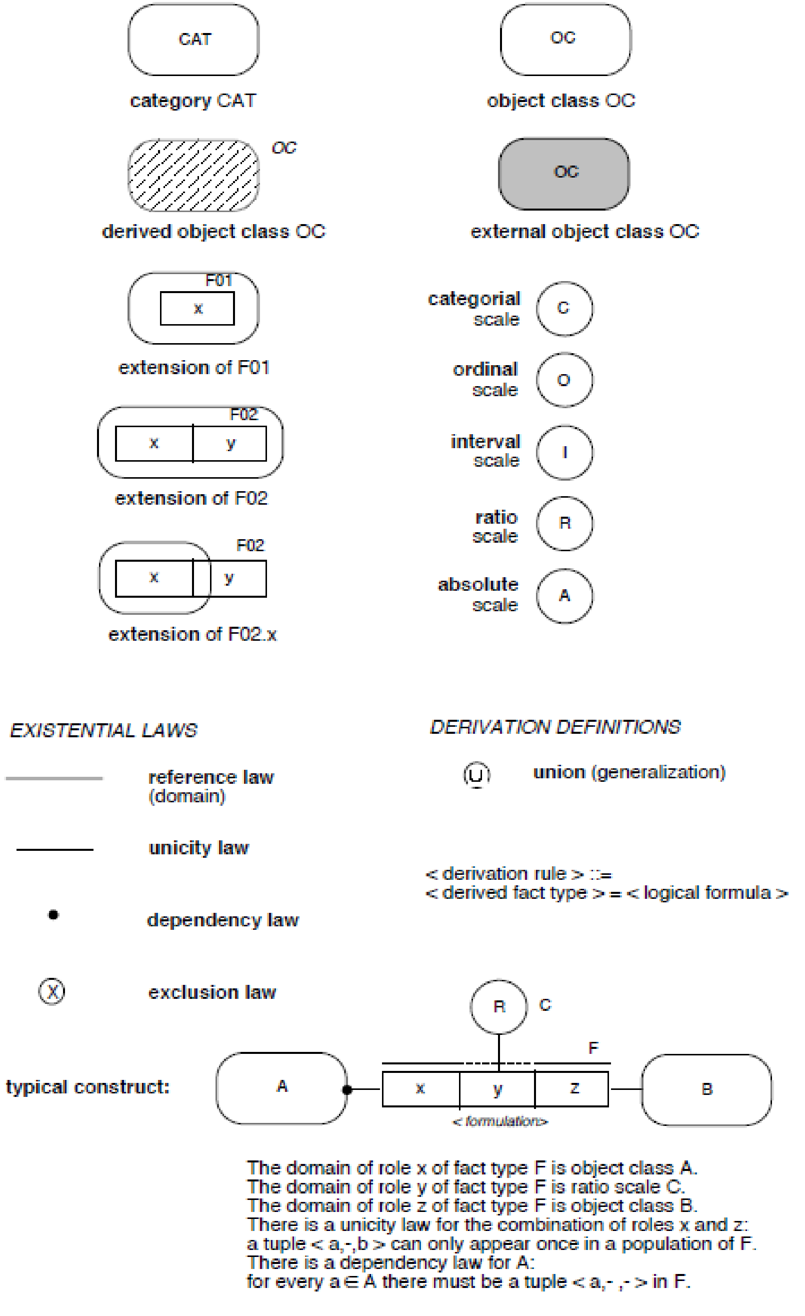

Figure 4.10 – Legend of the Object Fact Diagram (second part).

5

Related work – Why DEMO?

One of the questions that could be raised about this work is: why did we choose DEMO? The answer is simple. The reason why we picked up DEMO is because of the property of correctness and completeness it assures in its models. Due the Ψ-theory, DEMO methodology assures that the observed business transactions are structurally correct and complete. This quality gives DEMO properties that no other known modelling language can offer.

5.1 Business process modelling languages

This section introduces a brief description about some of the most popular modelling languages. Firstly we present the BPMN by describing its notation and language artefacts. Then, a short description of UML is made. The third and last modelling language that will be presented is the Archimate language. To finish this chapter, we present two related works in order to set some important comparisons between DEMO and BPMN, and then between DEMO and Archimate.

5.1.1 Business Process Modelling Notation (BPMN)

The Business Process Management Initiative (BPMI) has developed a standard Business Process Modelling Notation (BPMN). The primary goal of the BPMN effort was to provide a notation that is readily understandable by all business users, from the business analysts who create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and, finally, to the business people who will manage and monitor those processes (White, 2004). Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation (White, 2004).

BPMN defines a Business Process Diagram (BPD), which is based on a flowcharting technique tailored for creating graphical models of business process operations. A Business Process Model, then, is a network of graphical objects, which are activities (i.e., work) and the flow controls that define their order of performance (White, 2004).

A Business Process Diagram (BPD) is made up of a set of graphical elements. These elements enable the easy development of simple diagrams that will look familiar to most business analysts (e.g., a flowchart diagram). The elements were chosen to be distinguishable from each other and to utilize shapes that are familiar to most modellers (White, 2004). For example, activities are rectangles, and decisions are diamonds. It should be emphasized that one of the drivers for the development of BPMN is to create a simple mechanism for creating business process models, while at the same time being able to handle the complexity inherent to business processes (White, 2004). The approach taken to handle these two conflicting requirements was to organize the graphical aspects of the notation into specific categories. This provides a small set of notation categories so that the reader of a BPD can easily recognize the basic types of elements and understand the diagram (White, 2004). Within the basic categories of elements, additional variation and information can be added to support the requirements for complexity without dramatically changing the basic look-and-feel of the diagram (White, 2004). The four basic categories of elements are:

- ❖ Flow Objects
- ❖ Connecting Objects
- ❖ Swimlanes
- ❖ Artefacts

A BPD has a small set of (three) core elements, which are the Flow Objects, so that modellers do not have to learn and recognize a large number of different shapes. Many process modelling methodologies utilize the concept of *swimlanes* as a mechanism to organize activities into separate visual categories in order to illustrate different functional capabilities or responsibilities. BPMN supports swimlanes with two main constructs. Fig. 5.1 illustrates the core elements and the swimlanes used to represent a BPD.

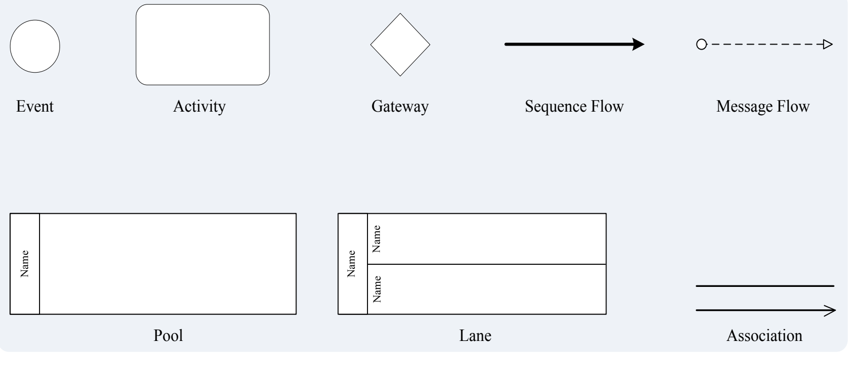

Figure 5.1 – Core elements of a BPM.

5.1.2 Unified Modelling Language (UML)

The Unified Modelling Language is the most-used modelling specification of the Object Management Group (OMG). UML allows users to model the business process, application structure, application behaviour, data structure, and architecture (Minoli, 2008). UML, along with the Meta Object Facility (MOF), also provides a foundation for OMG's Model-Driven Architecture, which unifies steps of development and integration from business modelling, through architectural and application modelling, to development, deployment, maintenance, and evolution (Minoli, 2008).

UML helps firms specify, visualize, and document models of software systems, including their structure and design. (Firms can use UML for business modelling and modelling of other nonsoftware systems too.) Utilizing any one of the large number of UML-based tools on the market, firms can analyze their application's requirements and design a solution that meets these requirements while representing the results using UML's 12 standard diagram types (Minoli, 2008). Within UML one can model most types of applications, running on any type and combination of hardware, operating system, programming language, and network (Minoli, 2008). We recommend Minoli, (2008, pp. 169 – 191) for more information about UML.

5.1.3 ArchiMate

ArchiMate is a language for modelling enterprise architectures in accordance with a meta model and a conceptual framework of modelling concepts, called the ArchiMate Framework (Ettema & Dietz, 2009). ArchiMate is based on the descriptive notion of architecture, which means that an enterprise architecture in ArchiMate corresponds to a conceptual model of the business processes in the enterprise (Ettema & Dietz, 2009). The ArchiMate Framework is exhibited in Fig. 5.2. Three architectural layers are distinguished, called the business layer, the application layer, and the technology layer (Ettema & Dietz, 2009). The idea behind this division is that the application layer provides services to the business layer, and that the technology layer provides services to the application layer. Moreover, the business layer is said to provide business services to the environment of the enterprise (Ettema & Dietz, 2009).

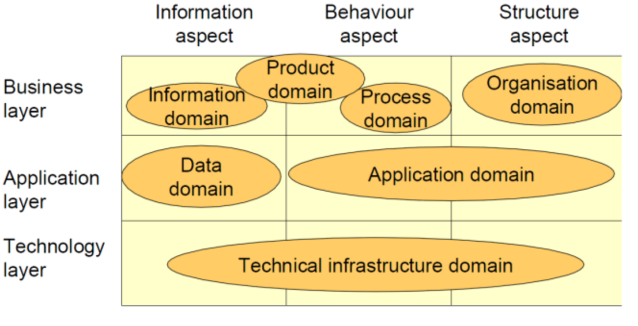

Figure 5.2 – The ArchiMate Framework.

On the horizontal axis, three major aspects are distinguished, called active structure, behaviour, and passive structure. The first two refer to generic system aspects, as e.g. identified by (Weinberg, 1975). The third aspect is related to the discipline of information system engineering. ArchiMate's perspective on organizations is highly influenced by this discipline. The concepts in Archimate (see Fig. 5.3) are structured in conformity with the framework of Fig. 5.2.

The conceptual structure therefore consists of active structural elements, of behavioural elements and of passive structural elements. Fig. 5.3 shows the Archimate concepts and main relationships. The concepts on the right hand side regard the active structure aspect. The concepts in the centre regard the behavioural or dynamic aspect, and the concepts on the left hand side regard the passive aspect.

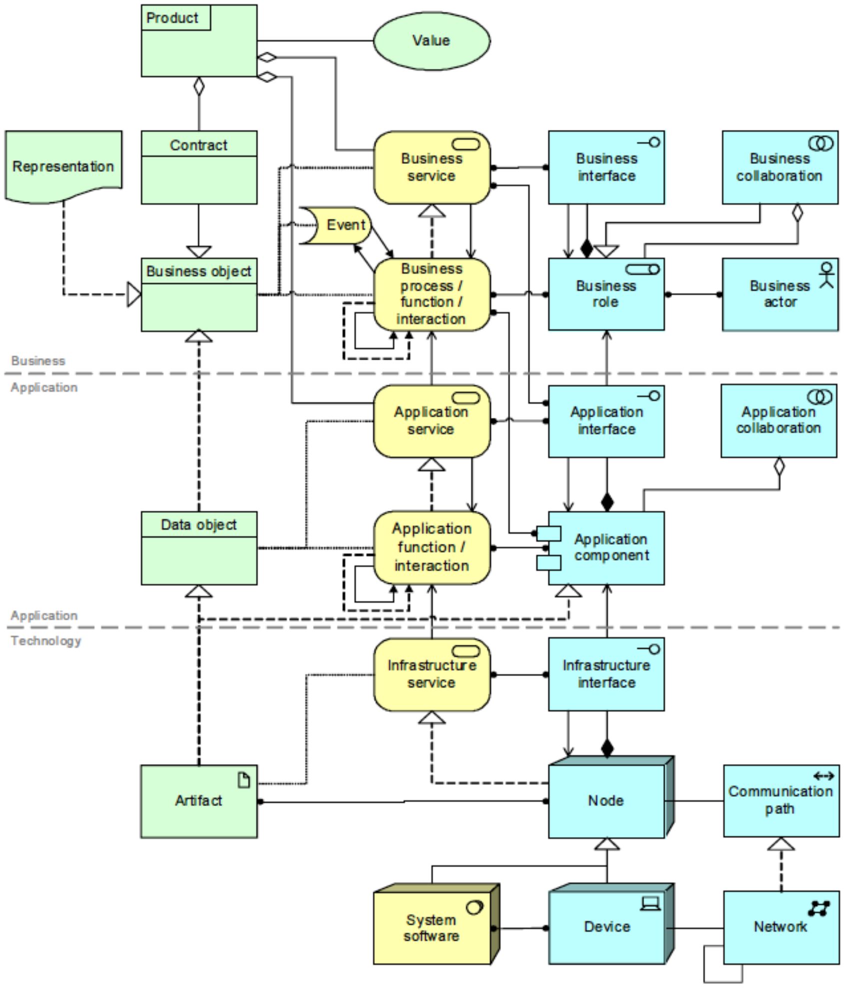

Figure 5.3 – ArchiMate concepts and main relationships.

5.2 DEMO versus BPMN – Theoretical comparison

Nuffel, Mulder, & Kervel, (2009) present us some conclusions about the enhancement of the formal foundation of BPMN by Enterprise Ontology (EO). We summarize here the conclusions of their work. For more information please see (Ettema & Dietz, 2009).

It can be stated that EO and DEMO have proven their strength for enterprise engineering. The main advantage is that a shared and formally correct conceptualization for all stakeholders has been established as the one and only right point of departure for any implementation.
Preliminary results of research by the authors show that the EO and the DEMO methodology deliver formally correct models which allow the construction of a DEMO automata, the core of Enterprise Ontology derived information systems. Moreover, DEMO ends where BPMN starts. As such, we gain the DEMO advantages and have a formal correct BPMN model.

Nuffel, Mulder, & Kervel, (2009) state that the design of enterprises, expressed by an ontology containing concepts of social interactions in its implementation includes an ontology for rational information systems, and an ontology for documental systems. However most ontologies are implicit regarding the concepts constructing the realities of their to be represented systems, such as the enterprise and information systems. Therefore it can be argued that documental and informational ontologies are subsets of an Enterprise ontology, leading to the conclusion that instead of investigating an one-to-one mapping, a many-to-many mapping has also to be further researched. The authors thus argue that Enterprise Ontology, combined with the DEMO methodology can provide a formal foundation to BPMN models. Using EO as underlying domain ontology results in explicitly specified BPMN models, mainly because EO delivers constraints to which BPMN constructs should adhere. Moreover, revising existing BPMN models with DEMO can be used to verify completeness and consistency of the modelled business processes. The main contribution of (Ettema & Dietz, 2009) is thus combining the rich representational aspects of BPMN with the formal correctness of DEMO.

To conclude, (Ettema & Dietz, 2009) present us a set of a consistency checks that can be performed regarding the communicative actions. These consistency checks could involve:

1. Checking the completeness of the BMPN diagram on the ontological level (are all communicative actions mapped?)
2. Check if each communicative action has a documental and/or informational implementation (this means that the customer call is mapped on the ontological as well as the documental level)
3. Checking the implementation of Actors to persons/departments
4. Checking the sequence of the communicative actions (are these conform the DEMO process model?)
5. Check the implications for the BPMN diagram of a redesign of the DEMO process model.

5.3 DEMO versus ArchiMate – Theoretical comparison

Ettema & Dietz, (2009) carried out a comparative evaluation of ArchiMate and DEMO, both theoretically and practically, i.e. on the basis of the analysis of the same case by each approach. In this section we present the conclusions achieved by the authors. For more information please see (Ettema & Dietz, 2009).

A thorough assessment of the strengths and weaknesses of ArchiMate and DEMO can only be performed on the basis of multiple real-life and real-size cases, taken from different areas. Nevertheless, some conclusions can certainly be justified already now.

The first conclusion is that ArchiMate and DEMO are hardly comparable, for several reasons. One is that ArchiMate is a second wave approach, whereas DEMO is a third wave approach. Another reason is that DEMO is founded in a rigorous and appropriate theory, whereas ArchiMate lacks such a foundation. Therefore, its semantics are basically undefined, which unavoidably leads to miscommunication among Archimate users. One would expect that having a rigorous semantic definition would be a prerequisite for an open standard.

A second conclusion regards the abstraction layers as distinguished by ArchiMate and DEMO. DEMO (in fact the Ψ-theory) makes a well defined distinction between three abstraction layers: the B-organization, the I-organization, and the D-organization. Only in the B-organization original new production facts are brought about (deciding, judging, manufacturing etc.), by which the enterprise world is changed. In the I-organization one computes, calculates, reasons; this does change the world. In the D-organization one stores, copies, transports etc., documents. Despite the fact that ArchiMate belongs to the second wave, it does not make a distinction between infological and datalogical issues in the business layer.

Although ArchiMate and DEMO are to a large extent incomparable they can usefully be combined (Ettema & Dietz, 2009). As a matter of fact, several studies have been carried out concerning the combination of DEMO with some second generation approach, since DEMO does not really cover the implementation of an organization. As one of the practical outcomes, a procedure has been developed for producing EPCs on the basis of DEMO models. In this way, the rigorous semantics of DEMO are so to speak enforced upon the EPC (Ettema & Dietz, 2009). Such a combination is also possible and beneficial for ArchiMate and DEMO (Ettema & Dietz, 2009).

6

Circular Method for Analysis and Compliance (CMAC)

This section describes a method that relies on DEMO methodology as a way to improve and verify the consistency and completeness of business process models (BPM). It encompasses two modelling approaches: a Bottom-up phase and a top-down phase. Both approaches of the method can be performed in a circular mode as illustrated in Fig. 6.1

6.1 The Bottom-up phase

As Described in table 6.1, the bottom-up phase is about three primary actions which we describe in this section.

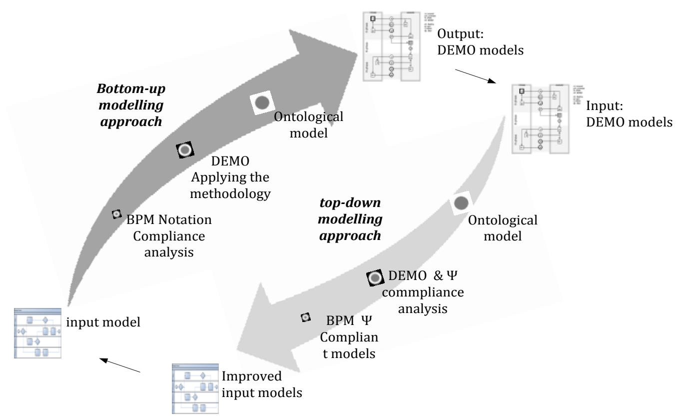

Figure 6.1 – The two approaches of the applied solution method.

6.1.1 Business Process Model Analysis

The primary action in the bottom-up phase is the analysis of the design elements or artefacts used to represent the business process model, which are typically:

- ❖ Decision points (e.g. gateways)
- ❖ Areas of responsibilities (swimlanes)
- ❖ Activity names (an activity is then evaluated according the PIF/CAP analysis as described in (Xie, Liu, & Dietz, 2005), in order to recognise acts of speech).

This analysis is finished when every acts of speech are properly identified and classified over the activity names described in the input model (after PIF/CAP analysis). Further, it allows us to distinguish every ontological acts of coordination performed by business actors. A more detailed discussion about the analysis of BMNs based on Nuffel, Mulder, & Kervel, (2009) and a set of hypothesis and propositions is now presented:

Because graph-oriented business process models like BPMN, for example, are composed of concepts, there must be some underlying domain ontology (Nuffel, Mulder, & Kervel, 2009). However, these models do not prescribe ontologies for business, information and data processes, and their results. Moreover, designers are free to choose their ontologies; and, in practice, the applied ontologies are mostly kept implicit (Nuffel, Mulder, & Kervel, 2009). Combining these two facts, it can be argued that Enterprise Ontology can be used as underlying domain ontology for graph-oriented/flowchart-based BPMs, exhibiting a high level of ontological appropriateness based on the arguments that the concepts (graphical elements) of these models are "similar" to the concepts of EO:

- ❖ The concept of an activity in flowchart-based BPM is "similar" to the concepts of either a performance or DEMO Production (P-) Fact; or some communication about the performance or DEMO Communication (C-) fact.
- ❖ The concept of a lane in flowchart-based BPM is "similar" to the concept of an Actor (Actor-Initiator or Actor-Executor).

To ground this hypothesis, both an evaluation of the EO ontological concepts, and a mapping to the primitives or concepts of flowchart-based BPMs have to be made. We start by evaluating the ontological concepts of Enterprise Ontology. As we saw in *chapter 4* EO is based on three axioms: the Operation Axiom, the Transaction Axiom and the Composition Axiom. The EO axioms are specified in schemes, i.e. formal graphical languages, and in natural language. A careful analysis is needed to identify all elementary and true propositions in natural language. An obvious requirement for completeness is that each entity in the EO axioms is identified (Nuffel, Mulder, & Kervel, 2009). The operation axiom, described in *chapter 4* states that there are Actors, performing some production or performance, and the actual production requires coordination between these Actors. There are relations between the elements of the set of Actors, the C-world containing coordination facts or C-facts, and the P-world containing P-facts (Dietz, 2006). Ontological appropriateness is supported by the observation that any performance or production involves an Actor(role), and some form of communication with another Actor(role) (Nuffel, Mulder, & Kervel, 2009). As such, five propositions can be derived:

- *Proposition 1*: There are identifiable Actors, fulfilling roles where an Actor refers to a role, and is not directly linked to an identifiable natural Actor (Nuffel, Mulder, & Kervel, 2009).
- *Proposition 2*: There are identifiable *Pfact*(s), representing some specific performance (Nuffel, Mulder, & Kervel, 2009).

- *Proposition 3*: There are *Cfact*(s), representing communication about a specific performance to be delivered (Nuffel, Mulder, & Kervel, 2009).
- *Proposition 4*: Each *Pfact* has a relation to one and only one unique *Cfact*, and vice versa (Nuffel, Mulder, & Kervel, 2009).
- *Proposition 5*: Every Actor role, except for the root Actor-initiator, has one and only one Actor-Executor relation to one and only one *Pfact*, while it may have multiple (0 .. n) Actor-Initiator relations to other (child) *Pfacts* (Nuffel, Mulder, & Kervel, 2009).

The Composition Axiom, described in *chapter 4*, states that *Pfacts* can mostly be ordered in a hierarchical parent-child tree structure, representing the fact that for a certain Pfact to be performed, first a number of *Child-Pfacts* representing performance or production components, must have been performed in a recursive way (Nuffel, Mulder, & Kervel, 2009). Ontological appropriateness is supported by the observation that many performances or productions require component performances or productions to become available first, as discussed in the composition axiom of *chapter 4*.
The Transaction Axiom, shown described in *chapter 4* in its most simple form of the basic transaction on the left and the standard transaction on the right, states that a transaction involves two Actor (roles), and follows a precisely specified pattern with states and state changes. The ontological appropriateness of the transaction axiom is strong: nested transactions up to unlimited nesting levels are computable and hence roll-back compatible (Nuffel, Mulder, & Kervel, 2009). The following predicates can thus be added to the earlier mentioned propositions:

- *Proposition 6* : There is a constructional decomposition type of relation between a specific Pfact and any number (0 .. n) of child Pfacts (Nuffel, Mulder, & Kervel, 2009).
- *Proposition 7* : For tangible Pfacts, all the child Pfact(s) have to be performed (Stated and Accepted) before the performance of the parent Pfact can start (Execution Phase) (Nuffel, Mulder, & Kervel, 2009).
- *Proposition 8* : An Actor with a Actor-Executor relation to a specific Pfact has an Actor-Initiator relation with each child Pfact (Nuffel, Mulder, & Kervel, 2009).
- *Proposition 9*: There is one and only one Pfact, the root Pfact, that has a relation to an Actor that has exclusively an Actor-Initiator relation to this root Pfact (Nuffel, Mulder, & Kervel, 2009).
- *Proposition 10* : There is at least one Pfact in a model without child Pfacts, i.e. a terminal Pfact, that has a relation to an Actor that has exclusively an Actor-Executor relation to this terminal Pfact (Nuffel, Mulder, & Kervel, 2009).

- *Proposition 11* : There is a set of eight attributes (Request, Promise, etc.) uniquely related to specific Cfact, that describes the current state of communication of that element about the performance of the related Pfact (Nuffel, Mulder, & Kervel, 2009).

Table 6.1 - Summary of the CMAC.

	Input	Output	Primary Actions
Bottom-up phase	Process model (e.g. BPMN, flowchart, workflow).	DEMO models (PSD).	1. BP source model analysis. 2. BP model transformation. 3. DEMO Model generation.
Top-down phase	DEMO models (PSD).	Revised DEMO models. Revised process model.	1. DEMO model compliance analysis. 2. DEMO model correction. 3. BP model generation. 4. Gap analysis.

6.1.2 Business Process Model Transformation

This analysis is about identifying and indicating every business actions in the input model that need to be capture in order to produce the DEMO model. It is performed by making the following distinctions over the activities described in the input model:

- ❖ Indicating (labelling) ontological acts (Coordination and production acts).
- ❖ Indicating (labelling) non ontological acts (Datalogical and infological acts).

Fig. 6.2 (A) illustrates the model produced from this transformation.

6.1.3 DEMO Model Generation

After performing the necessary transformations, the DEMO model is produced out of the identified coordination acts indicated and classified previously in the input model (business process model) as illustrated in Fig. 6.2 (B).

6.2 The Top-Down phase

In this phase the DEMO model produced as the output of the previous phase is now used as the input model. So the subsequent actions performed in this phase are described next.

6.2.1 DEMO Model Compliance Analysis

A compliance analysis is performed in order to verify if the produced DEMO model is compliant with the Ψ-theory described in chapter 4. In practice, it consists of verifying the Transaction and the composition axiom.

6.2.2 DEMO Model Correction

The DEMO model correction is about adding the steps or actions found missing when we generated the DEMO model from the BPM. These missing steps are the ontological acts that are not described explicitly in the analysed BPM. Through the Transaction Axiom every transaction phases (O-phase, E-phase, and R-phase) are revised and completed, and through the Composition Axiom the logical sequence of execution of the transactions is described. Fig. 6.2 (C) illustrates the resulting correct diagram.

6.2.3 Business Process Model Generation and Gap Analysis

After the correction of the produced DEMO model, we then perform a gap analysis in which we identify the steps or actions (ontological acts) in the DEMO model that are not reflected in the input BPM (these steps were coloured red in the diagram (C) of Fig. 6.2). This gap is filled by adding new activities in the BPM matching the newly added steps in the diagram (C) of Fig.6.2. Then the generated Ψ compliant BPM is, of course, illustrated in Fig. 6.2 (D).

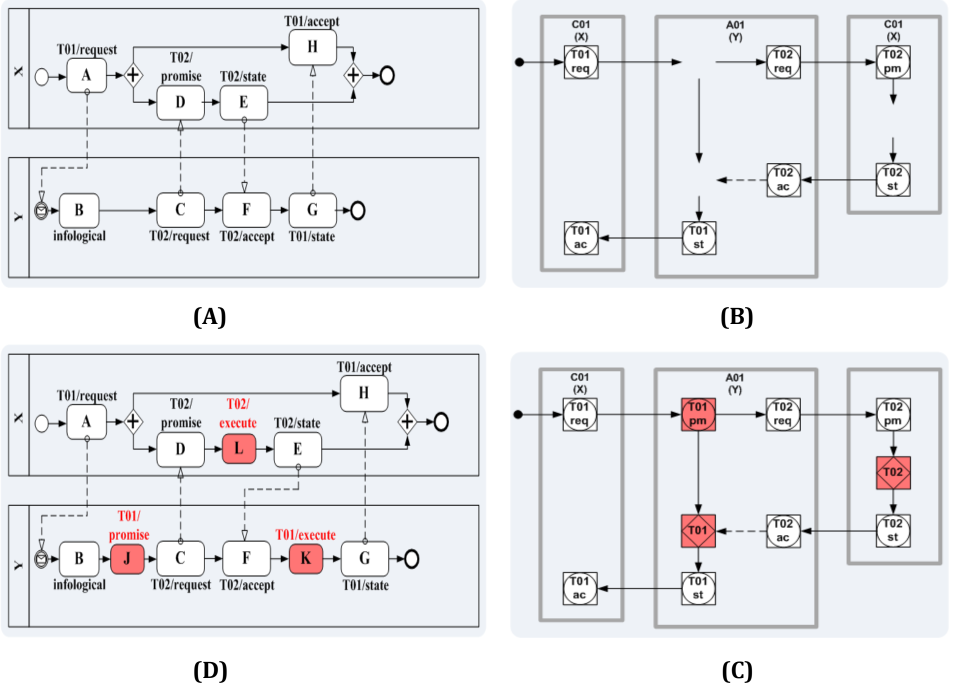

Figure 6.2 - Example of applying the CMAC over a BPMN model.

6.3 Formalizing contracts and responsibilities

In this section we demonstrate how to formalise a contract which describes different aspects of information about human interactions within organizations.

A contract is formalised through a template describing the responsibilities and obligations between two business actors within a specific business process transaction. These business actors are the contract initiator and the contract executor. A contract is based on the six aspects of the Zachman framework (ZF) and encompasses information like: the contract responsibilities, the transaction to be executed, the deadline in which a contract is expected to be fulfilled, the two parties taking action in the contract, the locations covered by the contract, and finally, the purpose of the contract.

6.3.1 The Zachman Framework

The ZF illustrated in Fig. 6.3 is widely used as an approach to structure and develop complex systems. It has been frequently adopted by IT organizations as a framework to identify and discipline the various perspectives involved in enterprise architecture. In practice, the ZF summarizes a collection of perspectives described in a two-dimensional matrix that has axes defined by type of stakeholders and aspects of the architecture (McGovern & Ambler, 2004, pp. 126-129).

	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Rel. Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location details	Event Details

Figure 6.3 – Zachman framework.

6.3.2 Relationship between Zachman and DEMO

An information system distinguishes itself from other software as it is developed to facilitate the operation of an organization; hence it reflects its strategies, plans, organizations, processes, marketing etc (Chen & Pooley, 2009.). Few methodologies are specifically designed for enterprise information systems, considering the fact that an information system distinguishes itself from other software as it is developed to facilitate the operation of an organization, hence it reflects the knowledge of the enterprise's structure, strategies, plans, organizations, people, activities, processes, resources, business rules, external relations etc (Chen & Pooley, 2009.). The complete computational representation of all such information can be called Enterprise Model or Enterprise Architecture (Chen & Pooley, 2009.). Research shows that a significant portion of requirements engineering activities is about acquiring and eliciting such information (Nuseibeh & Easterbrook, 2000), in other words, building enterprise architecture. These considerations are relevant enough to claim that enterprise ontology and DEMO can be of great value in the early stage of information systems development if we force the use of some logical and structured taxonomy for relating the concepts that describe the real world[18] to DEMO aspect models. This goal is accomplished

[18] The six enterprise aspects of the Zachman framework: *what, how, when, who, where and why.*

through the explicit mapping of DEMO on the Zachman enterprise aspects as we demonstrated next.

6.3.3 Contract Template – Mapping DEMO on Zachman Enterprise Aspects

A contract template is based on the six aspects of the Zachman framework (ZF) complemented with the information produced during the development of DEMO aspect models (see table 6.2). So the following table synthesises the information to be extracted from DEMO and mapped according the six aspects of the ZF to formalise a contract.

Table 6.2 – Mapping DEMO artefacts on ZF enterprise aspects.

Zachman Aspects	*DEMO source artefacts and contract information*
What (Contract Responsibilities)	*PSD diagram & IUT table – focus on the contract parties (initiator and executor), its obligations stated by the contract (C-acts and P-acts), and the information objects they use.*
How (Contract Transaction)	*PSD diagram & TRT table – focus on the transaction that needs to be performed in order to run the contract (the transaction type), as well as the type of result to be produced (the result type).*
When (Contract Deadline)	*PSD diagram & ATD diagram – focus on time aspects of the contract transaction.*
Who (Contract Parties)	*PSD diagram & ATD diagram – besides the involved actor roles, it also provides detailed information about delegation, which is: within each area of responsibility, who is responsible or delegated to perform what.*
(Where) Contract Locations	*PSD diagram & ATD diagram – it provides information about the location of the contract parties.*
(Why) Contract purpose	*PSD diagram & ATD diagram – describes what is expected to be achieved with the execution of the contract.*

It also means that a contract template focus mainly on the business layer of the organization, from which the DEMO models are produced. The greatest outcome brought by this fusion between ZF and DEMO can be its suitability for a top down approach for IT systems development. It provides the ontological aspects, and so, the essence of the organization, and at the same time, it offers architectural aspects of concern, so the system developer can use it to stay in touch with the goals to be achieved through the system being developed. Fig. 6.4 shows the conceptual or descriptive model of a contract template and a contract template itself. In the descriptive model of a contract each line indicates the ZF enterprise aspects and the columns indicate the DEMO sources of information to fill a fill a template. In the case study of chapter 7 we will a see a practical use of a contract template.

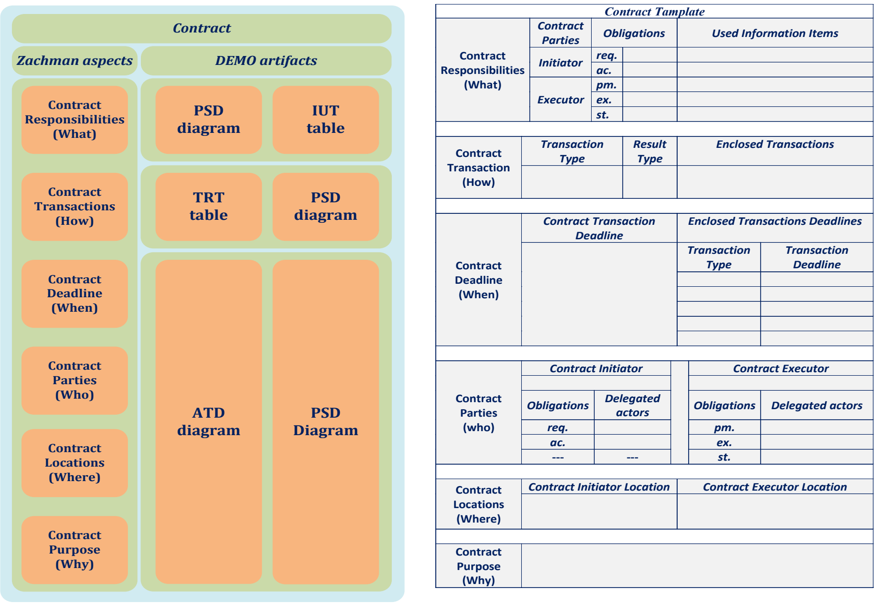

Figure 6.4 – Contract descriptive model (on left) and contract template (on right).

6.3.4 Chapter conclusions

About the method described in this chapter, as proposed by Ettema & Dietz, (2009) the following consistency checks are assured regarding communicative actions in a BPM being:

1. Checking the completeness of the BPM diagram on the ontological level (are all communicative actions mapped?)
2. Check if each communicative action has a documental and/or informational implementation (this means that the customer call is mapped on the ontological as well as the documental level)
3. Checking the implementation of Actors to persons/departments
4. Checking the sequence of the communicative actions (are these conform the DEMO process model?)

Check the implications for the BPM diagram of a redesign of the DEMO process model.

To finish this chapter we would like outline that the analysis described in this chapter is not a recipe, it involves some state of the art, knowledge, as well as, a good understanding of the business process we are evaluating, and most of the time it should be performed iteratively. It means, for example, that after performing one step we might find the need to go back and redo the previous step. In some situations, we could end up redoing every steps several times until get to the point where the input model is refined.

The execution of this evaluation will generally raise more questions than answers about the consistency and completeness of the input models, including the real meaning of some activities described in it. However, it is not bed news at all. Many of these questions might help to clarify responsibilities and even determine how it is being delegated and which levels of information access it implies. For example, in a customer service department where customer complaints are filed, every resolution plan implemented (produced) to attend a customer's complaint is ultimately a responsibility of that department, even if such responsibility is fully delegated or transferred to some other actor role. The transference of responsibilities through delegation might raise, in advance, questions about accountability and security, such as, which levels of access the delegated actor role should have towards the information involved? In cases where the delegated actor is not fulfilling its goals who should respond for that and who should be held accountable for that? In situations where business process reengineering is needed, the knowledge provided by the resulting ontological model can be useful to redesign the input business process diagram and from there, to suggest alterations at the business process structure.

7

Case Study – The Attorney General's Office

This chapter presents the thesis case study. It starts by presenting the Portuguese Attorney General's Office and describes its responsibilities, as well as, its organizational structure. Then the goals of the case study are described, followed by the applying of the solution method over the input model. Finally, at the end of the chapter, the results achieved are stated.

7.1 Case study goals

The goals of this case study were set according the research questions stated in section 3.7. So the following points indicate the main goals we intend to achieve in this chapter:

1. Apply DEMO to formalise contracts describing people's interactions within one of the BPM (*The Inquest Opening Process*) of the Portuguese Investigation Department (DIAP).
2. Apply DEMO to correct and improve one of the BPM (*The Inquest Opening Process*) of the Portuguese Investigation Department (DIAP).

7.2 Presentation

The Attorney General's Office of the Portuguese Republic, a hierarchical judiciary and autonomous public institution, has taken the decision to move forward towards integration and upgrade of its information systems. Such decision intends to improve workflow between all its departments and agents. The Attorney General's Office is responsible for representing the state in court of law, run prosecutions; guarantee the democratic legitimacy and the enforcement of the interests of law.

The competencies of the Attorney General's Office are divided into various levels, which include the criminal prosecution, under the responsibility of the criminal investigation department (DCIAP)[19], the promotion of legality, the defence of the interests of state, and the exercise of advisory functions. About its organizational structure, the Attorney General's Office has a pyramidal structure as shown by Fig. 7.1, where deputy prosecutors are represented in the bottom of the hierarchy, and on the top of it we have the attorney general.

[19] All acronyms refer to Portuguese institutional names, for more information about its detailed translation see table 1 in the appendix section (section 10).

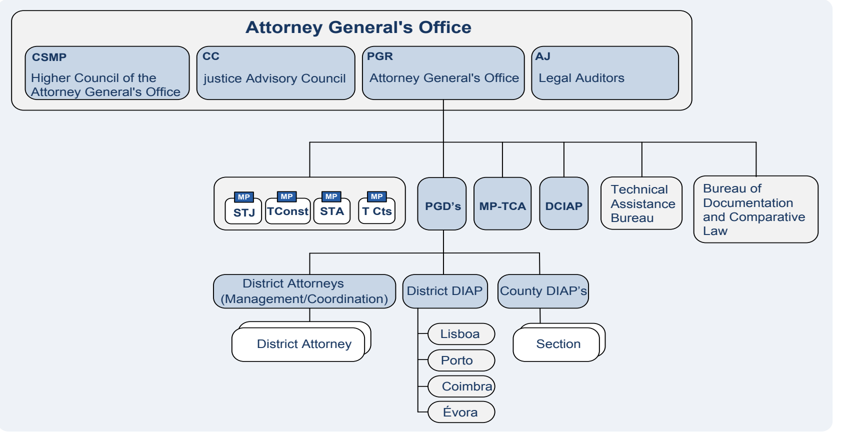

Figure 7.1 – Structure of the Attorney General's Office.

As illustrated in Fig. 7.1, the Attorney General's Office comprises the following four institutional entities: the Higher Council of the Attorney General's Office (CSMP), the Justice Advisory Council (CC) and Legal Auditors (AJ). Under the responsibility of the ATTORNEY GENERAL'S OFFICE are the Central Bureau of Investigation and Prosecution (DCIAP), the Bureau of Documentation and Comparative Law (GDDC), and the Technical Assistance Bureau (NAT). The DCIAP, also under the responsibility of the Portuguese Attorney General's Office, is responsible for coordinating and managing all criminal investigations and crime prevention activities. All District investigation departments (DIAP's) form an organized structure of the ATTORNEY GENERAL'S OFFICE, where each one has jurisdiction to run inquests and prosecute crimes committed inside its judicial district. Also under the responsibility of the ATTORNEY GENERAL'S OFFICE is the GDDC department, its mission is to collect, archive and disseminate information concerning community law, international law, and foreign law. At last, it is for the NAT department to ensure advice and technical consultancy to the ATTORNEY GENERAL'S OFFICE on economics, finance, and banking.

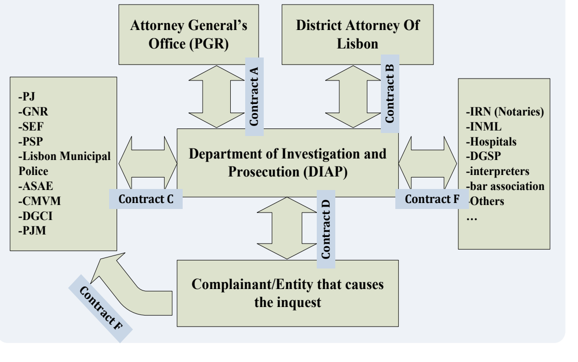

Figure 7.2 – DIAP and other external entities.

As one can see, in Fig. 7.2, some entities interacting with DIAP are illustrated. Interactions between these entities fall into the life cycle of the inquest. Fig. 7.2 shows those external entities responsible for causing the inquest, which can be ordinary citizens, OPCs (e.g. GNR, IS, PSP), or entities responsible for providing additional information during the life cycle of the inquest (e.g. hospitals, notaries, bar association, and others).

7.3 Case study Input business process

To perform this case study, we had access to documents describing the business processes of DIAP. This documentation was provided by the ITIJ[20],

In the context of the inquest, ten macro processes have been identified. These macro processes together form the life cycle and stages of the inquest. Besides covering information flows inside each department, they encompass information exchange and permanent interactions among the Lisbon investigation department (DIAP) and external entities. Due to space economy we'll only consider one of these macro processes for analysis, namely, the *Opening of the Inquest*, illustrated in Fig. 7.3.

7.4 Bottom-up phase

Considering the BPM of Fig. 7.3, which is our input model for the applying of the CMAC, we then describe the first three primary actions carried out.

7.4.1 Business Process Model Analysis

The primary action in the bottom-up phase is the analysis of the design elements or artefacts used in the input model of Fig.7.3. As one can see it is based on the following design elements:

- ❖ Decision points (gateways)
- ❖ Areas of responsibilities (swimlanes)
- ❖ Activities.

This analysis finished with the identification of all acts of speech, which was performed based on the PIF/CAP analysis over the names of the activities.

7.4.2 Business Process Model Transformation

By performing this analysis we were able to identify and indicate all business actions in the input model that need to be captured to produce the DEMO model. It was performed by making the following distinctions over the activities described in the input model:

- ❖ Indicate ontological acts (Coordination and production acts).
- ❖ Indicate non ontological acts (Datalogical and infological acts).

Fig. 7.4 illustrates the resulting BPM model produced in this transformation process.

7.4.3 DEMO Model Generation

After performing the transformations indicated above, then the Process Structure Diagram (PSD) of Fig. 7.6 was produced.

[20] Is the Information Systems Department of the Portuguese Attorney General's Office. For more information please visit: http://www.itij.mj.pt/PT/Paginas/Default.aspx

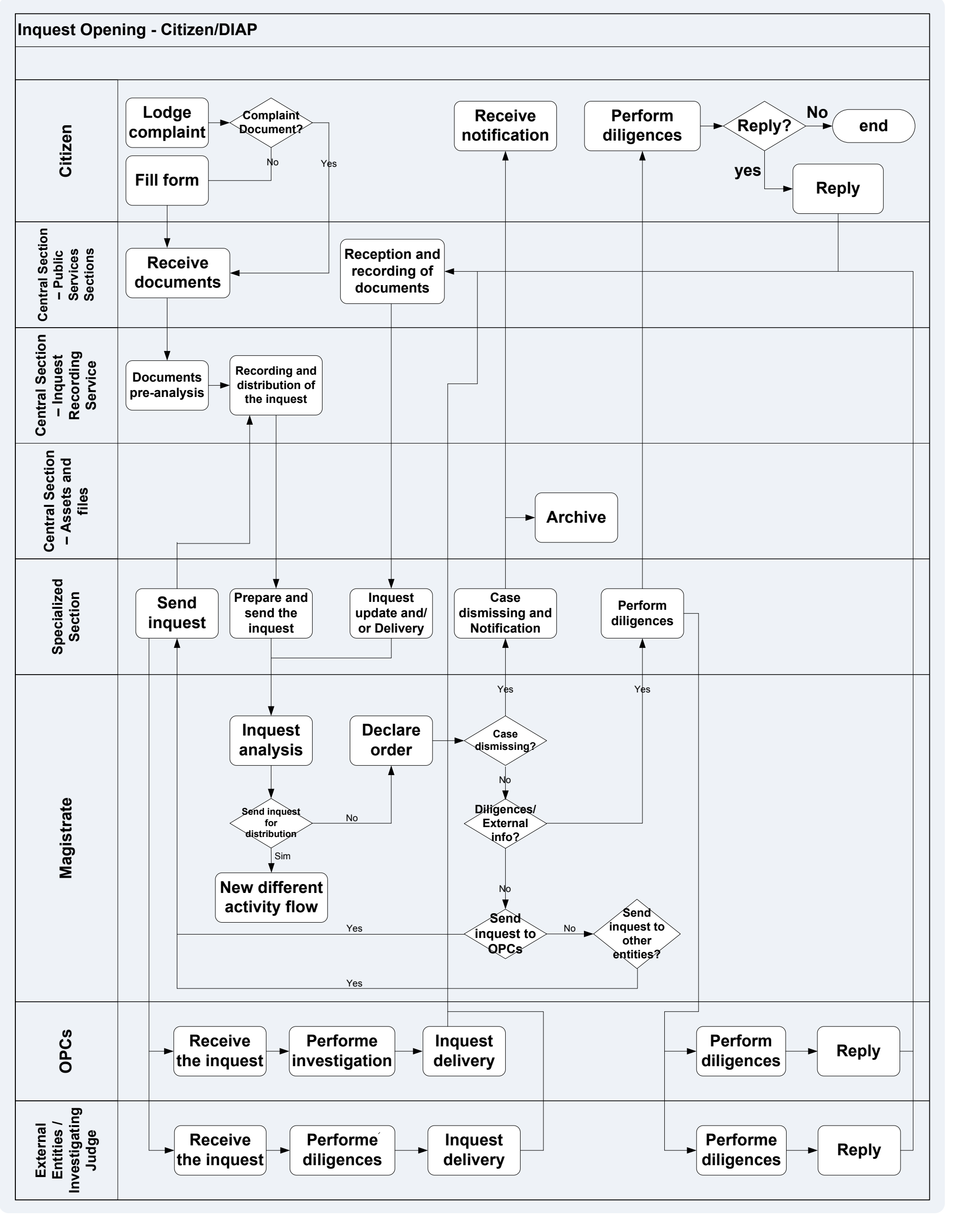

Figure 7.3 – Input Model: Inquest Opening Process.

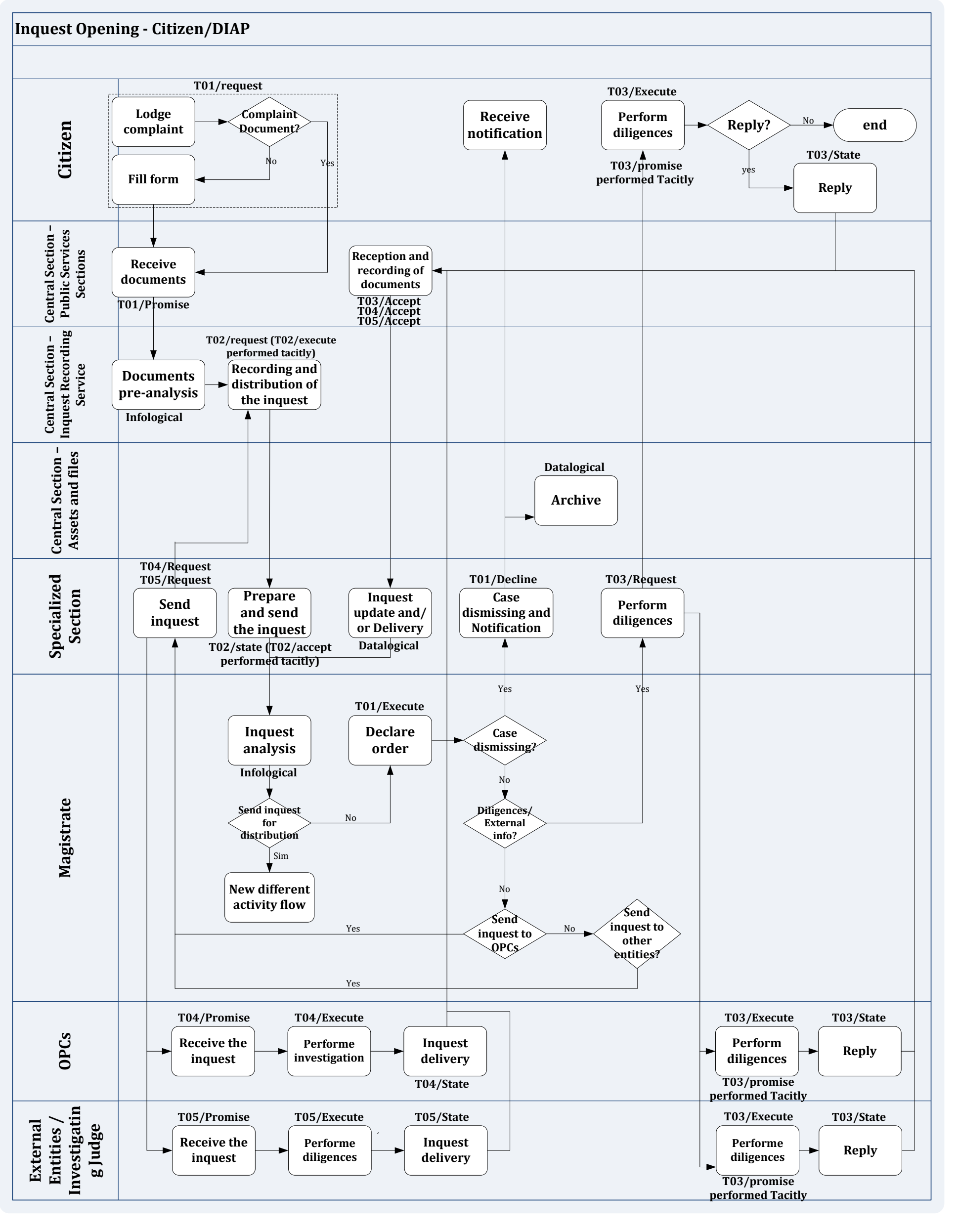

Figure 7.4 – The labelled input model: Inquest Opening Process

The PSD derived explicitly from the identified coordination acts indicated and classified previously in the input BPM of Fig. 7.4.

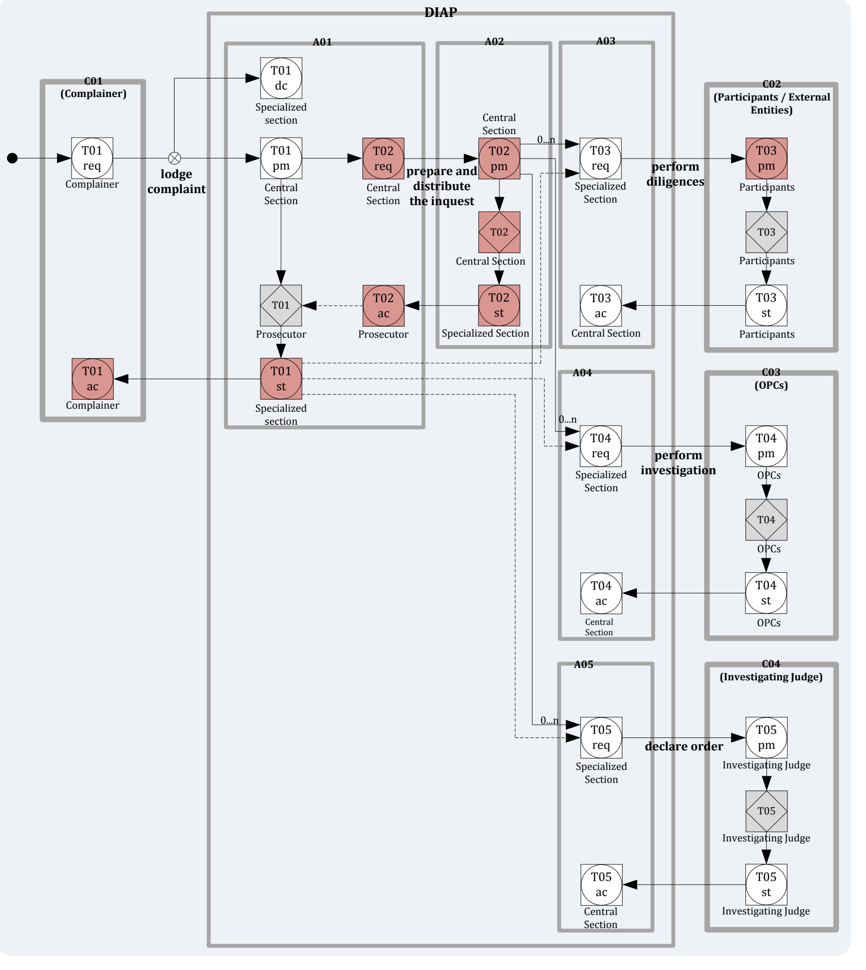

Figure 7.5 – The Process Structure Diagram (PSD).

For future production of *contract templates* we complete the Process Model (PM) of Fig. 7.5 by indicating the related Information Use Table (IUT).

Table 7.1 – Information Use Table (IUT).

object class, fact type , or result type	**process steps**
PERSON	T01/req
COMPLAINT	T01/req T02/req T02/pm
INQUEST	T02/st T02/ac T04/req T04/pm T05/req T05/pm
INVESTIGATING_JUADGE	T05/req
PROSECUTOR	T02/st
OPC	T04/req
SECTION	T02/st T03/st T04/st T05/st
EXTERNAL_ENTITIES	T03/ac
P is the complainer of C	T01/st
P is responsible for I	T02/st
I is submitted to investigating judge J	T05/ac
I is under investigation by OPC O	T04/ac

S is responsible for I	T03/st T04/st T05/st

The ontological *State Model* (SM) needed to produce the IUT of the PM is described by the *Object Fact Diagram* (OFD) illustrated in Fig. 7.6 and by the *Object Property List* (OPL) of table7.2.

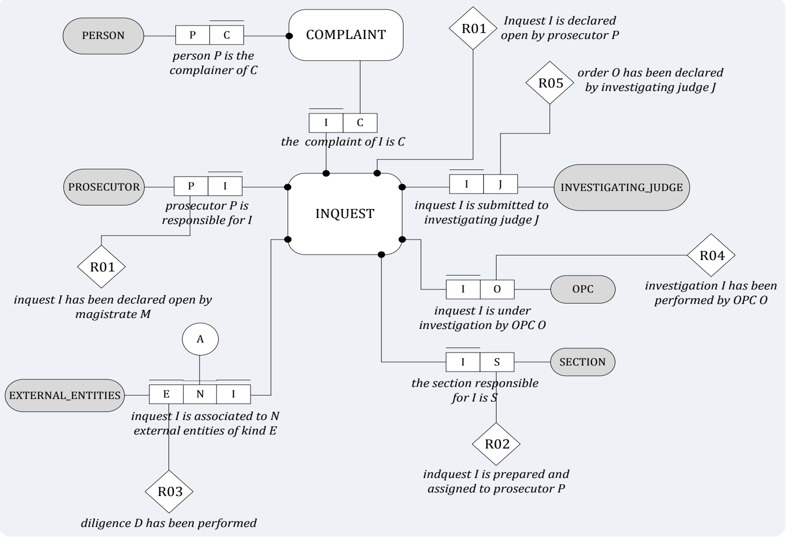

Figure 7.6 – The Object Fact diagram of (OFD) the Inquest Opening Process (IOP).

Table 7.2 – The Object Property List OPL.

property type	object class	scale
date_of_birth	PERSON	JULIAN DATE
age (*)	PERSON	NUMBER
current_address	PERSON	ADDRESS
person_id	PERSON	CITIZEN CARD NUMBER
type_of_complaint	COMPLAINT	TYPE OF COMPLAIN
date_of_complaint	COMPLAINT	JULIAN DATE
inquest_deadline	INQUEST	JULIAN DATE
inquest_id	INQUEST	INDENTIFICATION CODE
inquest_responsible	INQUEST	RESPONSIBLE
internal_id	PROSECUTOR	INTERNAL ID
internal_id	INVESTIGATING JUDGE	INTERNAL ID
police_agency	OPC	AGENCY
section_number	SECTION	NUMBER
entity_address	EXTERNAL ENTITIES	ADDRESS
entity_type	EXTERNAL ENTITIES	ENTITY_TYPE

To produce contract templates for the analysed BPM we also built the Construction Model (CM) formed by the Action Transaction Diagram (ATD) of Fig.7.7 and table 7.3. It indicates the actor roles, the boundaries of DIAP, as well as the transactions they participate in. The ATD does not tell us the specific steps delegated to each actor role as the PSD of Fig. 7.5 does, intead, it gives us the composition of DIAP, its environment, and its high level structure.

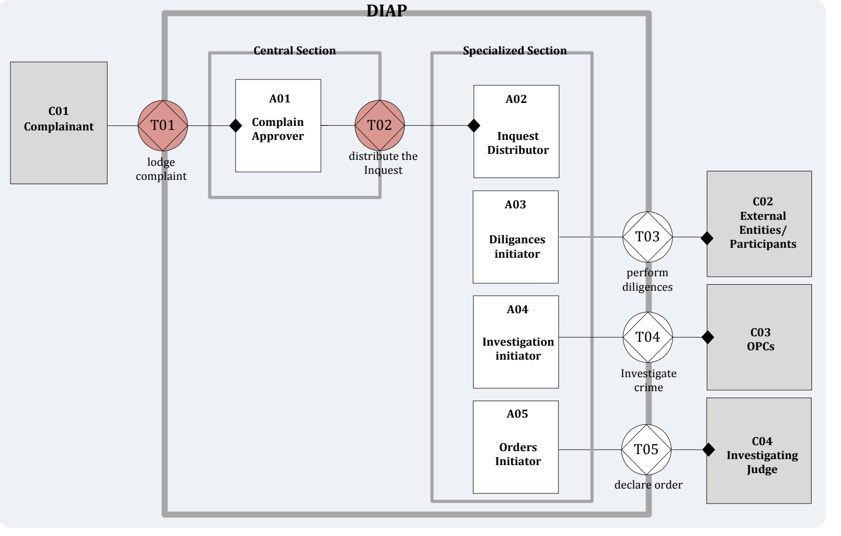

Figure 7.7 – The Action Transaction Diagram (ATD).

Table 7.3 is the Transaction Result Table (TRT) of the Inquest Opening Process. It indicates the results of each transaction illustrated in the ATD. For example, after the complete execution of transaction T01 the result produced is R01, which is, "*inquest I is declared open by prosecutor*". This information is quite relevant to formalise the contract as we will see in section 7.6

Table 7.3 – The Transaction Result Table (TRT).

transaction type	**result type**
T01 lodge complain	R01 inquest **I** is declared open by prosecutor **P**
T02 distribute the inquest	R02 inquest **I** is prepared and assigned to prosecutor **P**
T03 perform diligences	R03 diligence **D** has been performed
T04 perform investigation	R04 investigation I has been performed by OPC **O**
T05 declare order	R05 order **O** has been declared by investigating judge **J**

7.5 The Top-Down phase

In this phase, the PSD produced (Fig. 7.5) in the previous phase is now used as input. So the subsequent actions performed over that artefact are described next.

7.5.1 DEMO Model Compliance Analysis

As indicated earlier the compliance analysis is performed in order to verify if the produced DEMO model is compliant with the Ψ-theory as described in chapter 4. In particular, it consists of verifying the Transaction and the composition axioms.

7.5.2 DEMO Model Correction

We saw that the DEMO model correction phase is about adding the steps or actions found to be missing when generating the DEMO model from the BPM. These missing steps are the ontological acts that are not described explicitly in the analysed BPM (in this case the *Inquest Opening Process*). Through the Transaction Axiom every transaction phases (O-phase, E-phase, and R-phase) were revised and completed, and through the Composition Axiom the logical sequence of execution of the transactions were properly revised too. Due to space economy and because it took several iterations until we get the desired PSD, the added steps [21] were already coloured red in the diagram of Fig. 7.5.

7.5.3 Business Process Model Generation and Gap Analysis

After the correction of the produced DEMO model (the PSD), we then performed the gap analysis in which we identified the steps or actions (ontological acts) in the PSD that are not reflected in the input BPM (Fig. 7.8). Then, we generated the Ψ-compliant BPM as illustrated in Fig. 7.9.

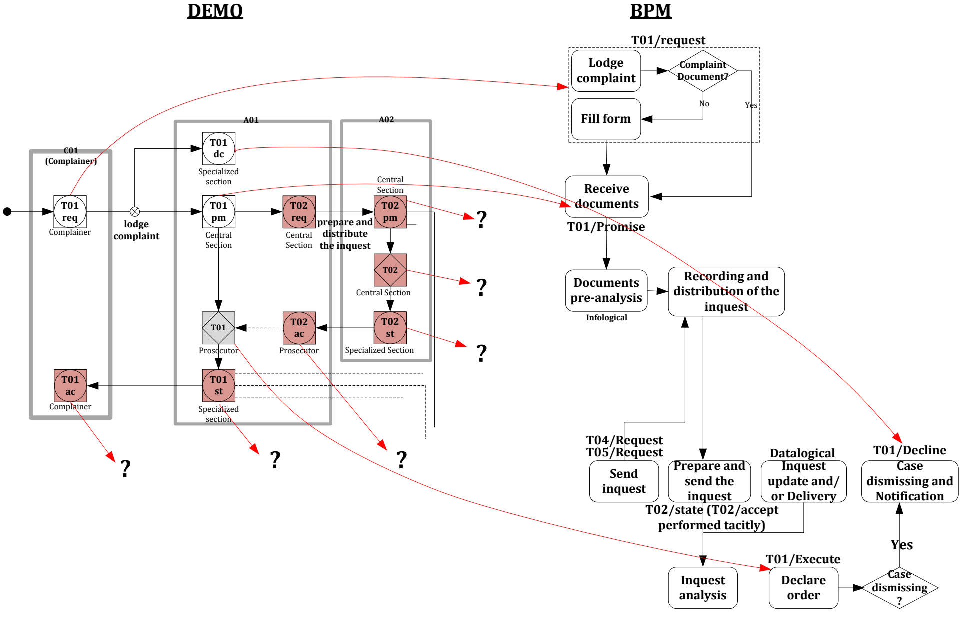

Figure 7.8 – Gap analysis: Unmatched process steps.

Fig. 7.8 illustrates two samples that we cut from the PSD diagram of Fig. 7.5 and from the BPM diagram of Fig. 7.4. It intends to demonstrate how we performed the gap analysis between both models. As shown, the steps coloured red with arrows pointing to question marks are the ones identified as not being matched in the BPM. Fig. 7.9 then presents the improved (Ψ-compliant) BPM where the new activities (also coloured red) derived from the DEMO model (through the gap analysis) were finally added to it.

The activities added in the BPM of Fig. 7.9 are normal procedures that are actually being carried out by people in the Portuguese Department of Investigation (DIAP). These activities were found missing in the original BPM of Fig. 7.3 but through this method we were able to indicated them and produce the desired Ψ compliant BPM.

[21] As indicated in Fig. 7.5 these missing steps are: T01/ac, T01/st, T02/req, T02/ac, T02/pm, T02/ex, T02/st, and T03/pm.

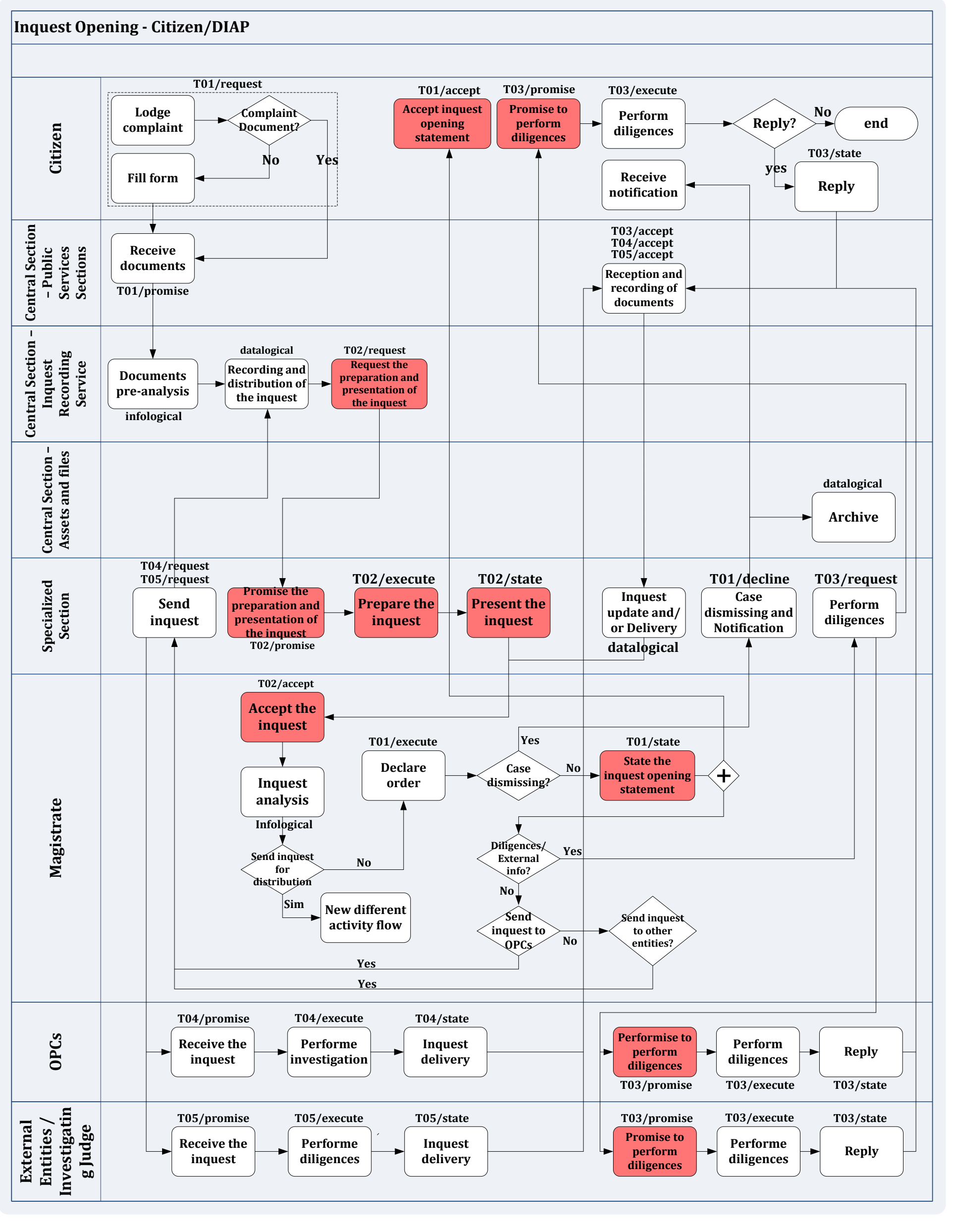

Figure 7.9 – Ψ compliant BPM.

7.6 Formalizing Contracts

In this section we indicate a contract that was formalized according to the method presented in *chapter 6*. The contract is described in table 7.4. The formalized contract is performed through transaction T01 of the ontological model developed in this chapter.

Table 7.4 indicates the responsibilities of the citizen who submits the complaint (the initiator of the contract) as well as the responsibilities of the executor of the contract which is the Central Section of the Investigation Department (DIAP). The contract transaction T01 can only be completed after the enclosed transaction T02 is finished. Which means the inquest is declared open by the prosecutor after T02 is completed. The contract also indicates the physical places where it will take place, and the purpose of the contract. It also specifies the obligations[22] of those involved in the contract. For this specific function of the Portuguese Investigation Department (to handle complaints) the *Central Section* promises the service, the prosecutor is responsible for producing the results and the *Specialized Section* has the obligation to state the results. Finally, the contract also indicates the information that people performing this contract need to have access to when fulfilling their obligations.

7.7 Identifying responsibility delegation contracts

Now, let's take a closer look again at the PSD of Fig. 7.5. As one can see delegation is being used resulting in authority being transferred among sections (from Central Section to Specialized sections) in the DIAP. Let's take, for example, actor role A03. Within the area of responsibility of A03 we can see two actions or process steps being performed, which are T03/req and T03/ac. Although we did not delegate those process steps to individual actor roles, which are more likely to be functionaries of the Investigation Department, and delegated them to internal departments (Central Section, and Specialized Sections of the DIAP), it is clear that T03/req and T03/ac are being performed separately by different actor roles – the Specialized Section and the Central Section, respectively. It raises a number of questions which can have impact on information systems design, and the way an organization operates. So, for the Investigation Department (DIAP), in particular, this issue raised the following questions:

- Every time a Specialized Section requests the performing of a diligence (T03/req), how is the Central Section informed that a new 'diligence result' is about to be stated (that a new T03/st is coming)? In other words, how does the Specialized Section inform the Central Section that it must accept (must perform a T03/ac) a result or a response that will be stated by a diligence performer (the performer of actor role C02)?

[22] Responsibilities acquired through (DEMO) delegation.

Table 7.4 – Formalized contract.

<table>
<tr><td colspan="7">Lodge Complaint Contract</td></tr>
<tr><td rowspan="6">Contract Responsibilities (What)</td><td>Contract Parties</td><td colspan="2">Obligations</td><td colspan="3">Used Information Items</td></tr>
<tr><td rowspan="2">Initiator</td><td>req.</td><td>lodge complaint</td><td colspan="3">PERSON</td></tr>
<tr><td>ac.</td><td>accept inquest opening statement</td><td colspan="3"></td></tr>
<tr><td rowspan="3">Executor</td><td>pm.</td><td>Receive documents</td><td colspan="3"></td></tr>
<tr><td>ex.</td><td>declare order</td><td colspan="3"></td></tr>
<tr><td>st.</td><td>state inquest opening t</td><td colspan="3">P is the complainer of C</td></tr>
<tr><td rowspan="2">Contract Transaction (How)</td><td colspan="2">Transaction Type</td><td>Result Type</td><td colspan="3">Enclosed Transactions</td></tr>
<tr><td>T01</td><td>lodge complaint</td><td>inquest I is declared open by prosecutor P</td><td>T02</td><td colspan="2">prepare and distribute the inquest</td></tr>
<tr><td rowspan="3">Contract Deadline (When)</td><td colspan="3">Contract Transaction Deadline</td><td colspan="3">Enclosed Transactions Deadlines</td></tr>
<tr><td colspan="3" rowspan="2"></td><td>Transaction Type</td><td colspan="2">Transaction Deadline</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td rowspan="6">Contract Parties (who)</td><td colspan="2">Contract Initiator</td><td rowspan="6"></td><td colspan="3">Contract Executor</td></tr>
<tr><td colspan="2">(C01) Complainer</td><td colspan="3">(A01) Central Section</td></tr>
<tr><td>Obligations</td><td>Delegated actors</td><td>Obligations</td><td colspan="2">Delegated actors</td></tr>
<tr><td>req.</td><td>Complainer</td><td>pm.</td><td colspan="2">Central Section</td></tr>
<tr><td>ac.</td><td>Complainer</td><td>ex.</td><td colspan="2">Prosecutor</td></tr>
<tr><td>---</td><td>---</td><td>st.</td><td colspan="2">Specialized Section</td></tr>
<tr><td rowspan="2">Contract Locations (Where)</td><td colspan="3">Contract Initiator Location</td><td colspan="3">Contract Executor Location</td></tr>
<tr><td colspan="3">(C01) Complainer</td><td colspan="3">(A01) Central Section</td></tr>
<tr><td>Contract Purpose (Why)</td><td colspan="6">To accept complaints and formalise the opening of the inquest.</td></tr>
</table>

- ❖ In the context of the inquest, what constraints are imposed, which is, what is the level of access the Central Section should have over the information received by the diligence performer?
- ❖ During the act of acceptance (T03/ac), what other responsibilities are passed to the Central Section? Will its responsibilities be merely datalogical (concerned about the form of the information) or infological (concerned about the content of the information)?

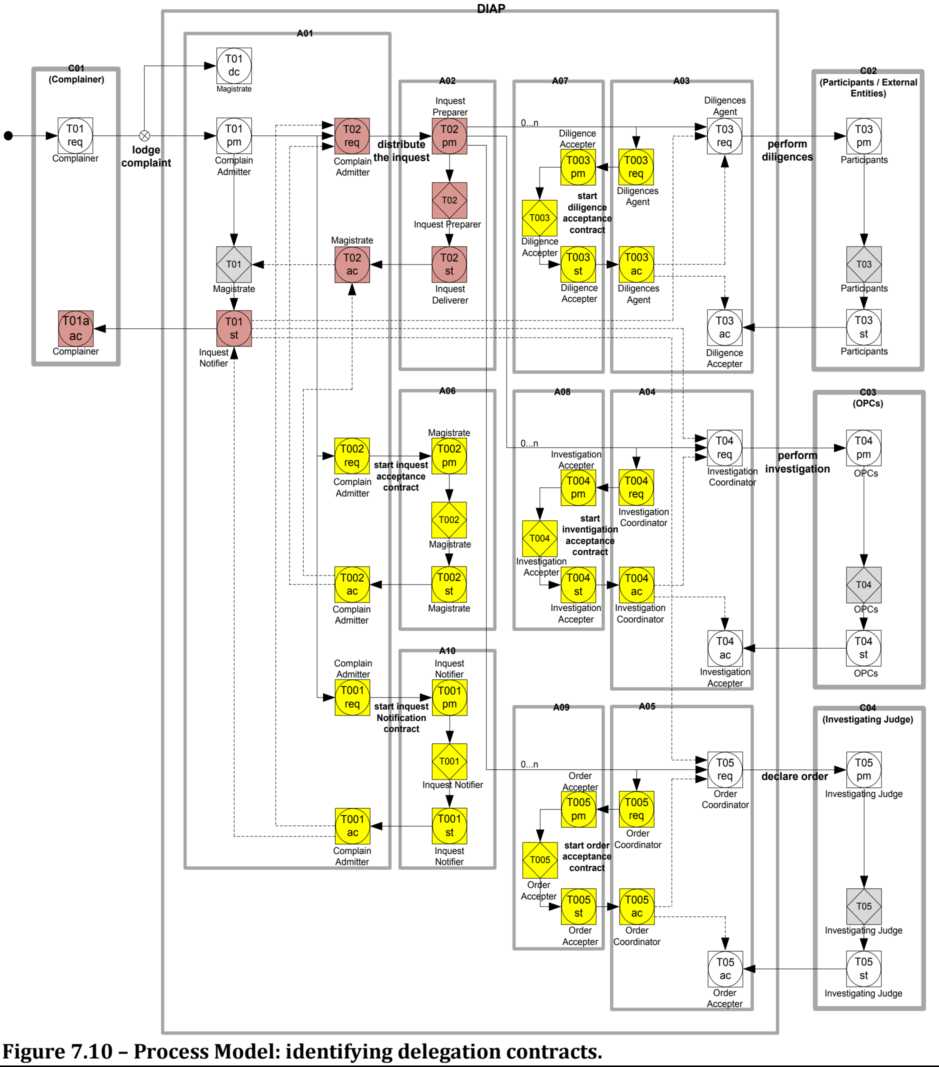

Figure 7.10 – Process Model: identifying delegation contracts.

The questions raised above are similar to the ones that could be raised for transaction T01, T02, T04 and T05. To answer them, we then proposed changes on the *as-is* process model of Fig. 7.5. These changes are depicted in the diagram of Fig. 7.10.

Firstly, we became aware, for example, that if someone in the Investigation Department (DIAP) initiates transaction T03 (*perform diligences*), by performing T03/req, then, simultaneously, he or she has to activate, somehow, some other 'mechanism' or procedure, in fact, a new transaction, which the aim is to inform the 'message receiver' – the performer or delegated entity of T03/ac – that an instance of the transaction T03 has been initiated. So the performer of T03/ac now '*knows*' about the existence of a new instance of T03 and should be prepared to deal with it. In other words, there must be some kind of contract between the one delegated to perform T03/req, and the one delegated to perform T03/ac. Particularly, this contract is fulfilled by transaction T003, coloured yellow in the diagrams of Fig. 7.10 and 7.11. In fact, all transactions coloured yellow in these diagrams (T002, T003, T004, and T005) represent what we call '*delegation contracts*', in which, two delegated entities (the performer of a '*request*' and the performer of an '*accept*' or '*state*' process step – see Fig. 7.11) set an agreement about performing distinct steps of the same transaction, under the same *area of responsibility*. That agreement is supposed to establish a set of rules, procedures, restraints, deadlines and information rights about the content of the information being exchanged in the context of the inquest.

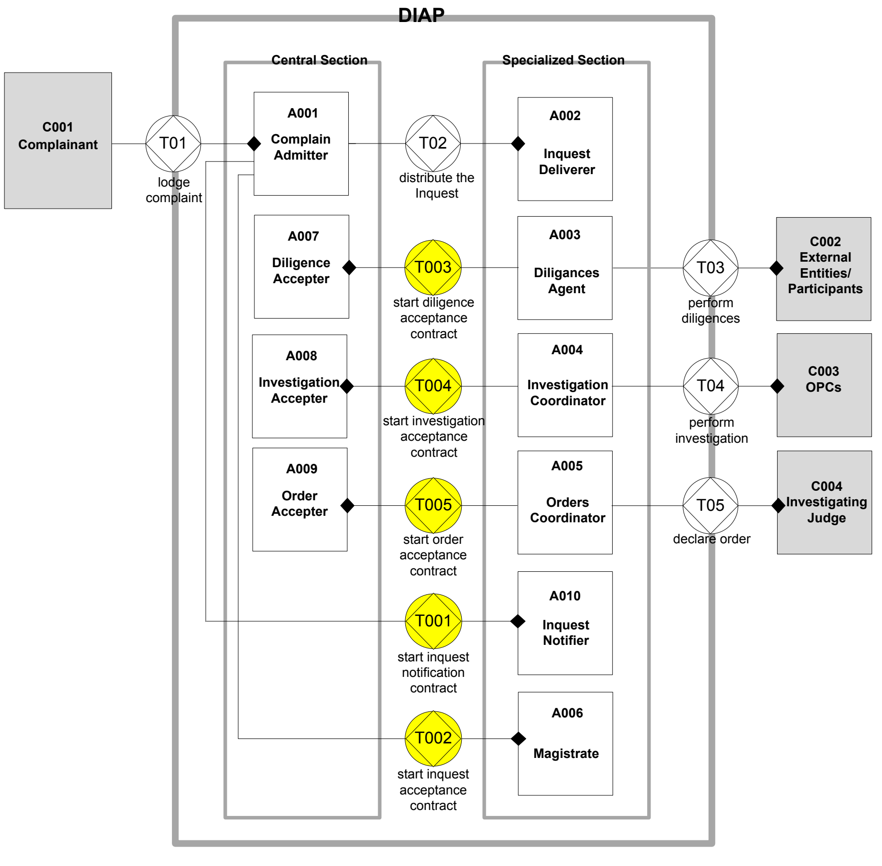

Figure 7.11 – Construction Model: identifying delegation contracts.

7.8 Results assessment

According to analysis performed, this case study provided the following contributions:

1. To detect inconsistencies in the analysed input models of the DIAP and suggest its refinement.
2. To detect implicit assignment and sharing of responsibilities among the involved actor roles in the input models.
3. To foresee the existence of implicit contracts (responsibility delegation contracts) regulating the sharing of responsibilities among the involved actor roles.
4. To identify and describe (through contract templates) contracts and responsibilities between the DIAP and other external entities indicated in the analysed input model.

8

Conclusions

This chapter summarizes the concepts introduced in this work, the major results produced, and its contributions. The thesis problem, the questions and goals stated on chapters 2 and 3 respectively are now evaluated here and confronted with the results and the objectives achieved. We'll also present a brief discussion about the thesis artefacts produced, highlighting its relevant aspects and contributions. To end, we will talk about future work considering the results produced this research and its contributions.

8.1.1 Thesis concepts

In this thesis wok we introduced the *Circular Method for Analysis and Compliance* (CMAC), described in chapter 6. The CMAC is based on two approaches namely:

i. The *bottom-up phase*.
ii. *The top-down phase*.

We also introduced the concepts of *contract* and *contract template*. A contract template expresses a contract within a formal document indicating different organizational aspects covered by an interaction between two business actors. A contract also clarifies the responsibilities of both business actors.

8.1.2 Thesis evaluation

The thesis evaluation addresses two aspects: the application scenario and the thesis research questions. In terms of the application scenario we addressed the following problems:

Application scenario (DIAP):

Problem one:
In the context of the Inquest, how do we formalise contracts describing human interactions?
Problem two:
In the context of the Inquest, how can we identify inconsistencies and improve the business process models?
Proposed solution:

The solution was demonstrating by applying the CMAC to improve one of the BPM describing the Opening of the Inquest in the Investigation Department (DIAP). The method proved to be efficient and allowed us to point and correct deficiencies found in the analysed BPM.

Research questions:

Thesis research question one:

Can we use DEMO to identify inconsistencies and improve graph-oriented business process models?

Thesis research question two:

Is DEMO suitable to formalise contracts describing human interactions within a business process?

Thesis answer:

Through the thesis cased study in chapter 7 we demonstrated that DEMO is suitable not only to clarify explicit responsibilities but also implicit ones. From the DEMO models produced by applying the CMAC we created the means to express those responsibilities formally (through contracts templates), and additionally, we managed to indicate in the ontological model every implicit contracts (which we designated as responsibility delegation contracts – see section 7.7)

8.1.3 Future work

As future work we propose the following topics for research:

1. *Automated tool to translate flowchart-based models to DEMO models and DEMO models to flowchart-based models*: by exploring the CMAC, we propose as future work the development of a tool capable of transforming correctly a flowchart-based model to its correspondent DEMO model and vice versa. Such a tool would speed up the analysis work performed to study and refine a BPM.
2. Automated tool to analyse the conformance of a flowchart-based model with the Ψ-theory: such a tool could be interesting as in terms of verifying the completeness and coherence of models such as BPMN diagrams.

Bibliography

Aalst, M. V., & Arthur, H. H. (2000). Verification of Workflow Task Structures: A petri-net-based Approach. *Information Systems , 25* (1).

Barjis, J. (2007). Automatic business process analysis and simulation based on DEMO. *Enterprise Information Systems , Volume 1 Issue 4*.

Benveniste, A., Caillaud, B. C., Ferrari, A., Mangeruca, L., Passerone, R., & Sofronis, C. (2008). Multiple Viewpoint Contract-Based Specification and Design. *Formal Methods for Components and Objects* , pp. 200 - 225.

BPMI.ORG. (n.d.). Business Process Management Initiative - http://www.bpmi.org.

Chen, Z., & Pooley, R. (2009.). Rediscovering Zachman Framework Using Ontology from a Requirement Engineering Perspective. *Computer Software and Applications Conference* (pp. 3 - 8). Seattle, WA : 33rd Annual IEEE International .

Christensen, Erik; Curbera, Francisco; Meredith, Greg; Weerawarana, Sanjiva. (2001, March 15). *Web Services Description Language (WSDL) 1.1*. Retrieved May 05, 2011, from W3C: http://www.w3.org/TR/wsdl

Decker, G., & Barros, A. (2008). Interaction Modeling Using BPMN. *BUSINESS PROCESS MANAGEMENT WORKSHOPS - Lecture Notes in Computer Science , Volume 4928/2008*, pp. 208-219.

Dewsbury, G., & Dobson, J. (2007). Responsibility and dependable systems. Springer.

Dietz, J. (2006). *Enterprise Ontology - Theory and Methodology.* (Springer, Ed.)

Dijkmana, R. M., Dumas, M., & Ouyang, C. (2008, November). Semantics and analysis of business process models in BPMN. *Information and Software Technology , Volume 50, Issue 12*, pp. 1281-1294.

Ettema, R., & Dietz, J. L. (2009). ArchiMate and DEMO – Mates to Date? *Volume 34, Part 4, 172-186* . Springer.

Gordijn, J., Akkermans, H., & Vliet, H. v. (2000). Business Modelling Is Not Process Modelling. *CONCEPTUAL MODELING FOR E-BUSINESS AND THE WEB - Lecture Notes in Computer Science , Volume 1921/2000*, pp. 40-51.

Group, O. M. (1997-2011). *Business Process Management Initiative*. Retrieved from http://www.bpmn.org/

Henrique, M. (2010). *Enterprise Governance and DEMO - Towards a reference method to guide the enterprise dynamics by addressing DEMO's competence, authority and responsibility notions.*

Hevner, A. (2004). Design science in information systems research. *MIS Quarterly , Volume 28*, pp. 75-107.

Kabilan, V. (2005). Contract Workflow Model Patterns Using BPMN. *10 th International Workshop on Exploring Modeling Methods in Systems Analysis and Design.* CiteSeerX Scientific Literature Digital Library.

Knowledge Based Systems, I. (1981). *IDEF0 method report*. Retrieved from http://www.idef.com/Complete Reports/idef0

Lankhorst, M. (2005). *Enterprise Architecture at Work - Modelling, Communication, and Analysis* (First Edition ed.). The Netherlands: Springer-Verlag Berlin Heidelberg.

Liu, K. (2000). *Semiotics in Information Systems Engineering.* Cambridge, United Kingdom: Cambridge University Press.

McGovern, J., & Ambler, S. W. (2004). *A practical guide to enterprise architecture.* Prantice Hall.

Minoli, D. (2008). *Enterprise Architecture A to Z, Frameworks, Business Process Modeling, SOA, and Infrastructure Technology.* (C. press, Ed.)

Nuffel, D. V., Mulder, H., & Kervel, S. V. (2009). Enhancing the Formal Foundations of BPMN by Enterprise Ontology. *ADVANCES IN ENTERPRISE ENGINEERING III - Lecture Notes in Business Information Processing , Volume 34, Part 3*, pp. 115-129.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: a roadmap. *Proceedings of the Conference on The Future of Software Engineering* (pp. 35-46). New York, NY, USA: ACM.

Recker, J., Indulska, M., Rosemann, M., & Green, P. (2006). How Good is BPMN Really? Insights from Theory and Practice. pp. 1582–1593.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modelling Language Reference.* Reading, Massachusetts: AddisonWesley Longmann, Inc.

Shishkov, B., Xie, Z., Liu, K., & Dietz, J. (2005). Identifying Business Process Patterns Using Semantic Analysis. (Springer, Ed.) *VIRTUAL, DISTRIBUTED AND FLEXIBLE ORGANISATIONS , III*, pp. 303-305.

Strens, R., & Dobson, J. (1993, Sept). How responsibility modelling leads to security requirements. (16th National Computer Security Conference). New Security Paradigms Workshop.

van Hee, K. M. (1994). *Informations Systems Engineering - A formal approach.* Cambridge: Cambridge University.

Weinberg, G. M. (1975). An Introduction to General Systems Thinking.

White, S. A. (2004, July). Introduction to BPMN.

Xie, Z., Liu, K., & Dietz, J. (2005). Identifying Business Process Patterns Using Semantic Analysis. *VIRTUAL, DISTRIBUTED AND FLEXIBLE ORGANISATIONS , III*, pp. 303-305.

Zdravkovic, J., & Kabilan, V. (2005). Enabling Business Process Interoperability Using Contract Workflow Models. *ON THE MOVE TO MEANINGFUL INTERNET SYSTEMS 2005: COOPIS, DOA, AND ODBASE , Volume 3760/2005*, pp. 77-93.

Zimmermann, O., Schlimm, N., Waller, G., & Pestel, M. (2005). Analysis and Design Techniques for Service-Oriented. *CiteSeerX* .