FEDERAL UNIVERSITY OF THE STATE OF RIO DE JANEIRO
CENTER OF EXACT SCIENCES AND TECHNOLOGY
SCHOOL OF APPLIED INFORMATICS

AO-BPM 2.0: Improving Aspect Oriented Business Process Modeling Notation

Luiz Paulo Carvalho da Silva

Mentor
Flávia Maria Santoro, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
DECEMBER 2016

AO-BPM 2.0: Improving Aspect Oriented Business Process Modeling Notation

Luiz Paulo Carvalho da Silva

Undergraduate thesis and graduation project presented at the School of Applied Informatics at Federal University of the State of Rio de Janeiro (UNIRIO) in order to obtain the title of Bachelor in Information Systems.

Approved by:

_____

Flávia Maria Santoro D.Sc. (UNIRIO)

_____

Claudia Cappelli D.Sc. (UNIRIO)

_____

Leonardo Guerreiro Azevedo D.Sc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.
DECEMBER 2016

Ποταμοῖς τοῖς αὐτοῖς ἐμβαίνομέν τε καὶ οὐκ ἐμβαίνομεν, εἶμέν τε καὶ οὐκ εἶμεν.

"*We both step and do not step in the same rivers. We are and are not.*"

Heraclitus

"*Doubt is not a pleasant condition, but certainty is absurd.*"

Voltaire

"*Plans are useless but planning is indispensable.*"

Dwight D. Eisenhower

"*The limits of my language mean the limits of my world.*"

Ludwig Wittgenstein

Acknowledgements

As a good exact sciences student, acknowledgments will come in unnumbered list:

- Humbly to the father of my head, Babalú-Ayé, and the mother of my head, Yemoja, who led me spiritually in every journey of my life. To all the spiritual guides who were by my side, in special to Mariazinha do Jardim who encouraged me in my access exam, spiritually helping me.
- My relatives, who stood beside me and supported me, encouraged and extended their hands when I needed them most.
- To my course friends, who were, at various times, the reason for my continuation in the course and who gave me strength, company, affection and attention. Listing, but not limited to these: Igor Balteiro, Victor Farias, Luis Felipe Sobral, Natália Costa, Matheus Costa, Carolina Tozzi, Victor Hugo Brandão, Lucas Lopes, Daniel Villaça, Thiago Marçal, Diogo Ferreira, Bernardo Gouveia, Amanda Rodrigues, Ygor Bessa, etc.
- To my teachers, who helped build what I became with dedication and commitment, not just intellectually.
- To my friend and companion Jonatham Petzold, who endured my unbearably academic personality and gave me books that were essential to the development of this work.
- To Mr. Rogério Oliveira, who gave me with my first imported book in Business Process Management.
- To my beloved and dear director Morganna Diniz, my coordinator Geiza Hamazaki and the Dean of my center, Luiz Amâncio.
- To professor Claudia Cappelli, who practically co-mentored me through graduation and introduced me to the Transparency area. We sing the blues sometimes and do not always understand each other (since we have a very similar personality), but we had done some phenomenal work.
- Lastly and most importantly, my dearest teacher and mentor Flávia Maria Santoro, who introduced me to the Business Process Management area. I have no words to express my gratitude and feelings. From the bottom of my heart, thank you so much.

RESUMO

Negócios e sistemas são graficamente representados para que sejam efetivamente compreendidos, e nestes as suas atividades. Diversas linguagens e notações propõem meios para esta representação, com suas próprias características sintáticas, semânticas e pragmáticas. A *Business Process Model and Notation* permite a construção de modelos de atividades mais compreensíveis e analisáveis aos seus interessados, não se preocupando com interesses espalhados por eles, com a relevância de suas atividades ou com a diferenciação do escopo, representando então blocos monolíticos de informação. A Orientação a Aspectos buscou resolver esta questão inicialmente na programação, servindo como base para elaboração de linguagens para sistemas orientados a processos e posteriormente para a Modelagem de Processos de Negócios. Este trabalho apresenta uma comparação e avaliação de notações para modelagem de processos de negócio utilizando o paradigma da orientação a aspectos, análise aprofundada de uma delas (AO-BPM 2.0) e sua operacionalização em um cenário real.

Keywords in Portuguese: Gestão de Processos de Negócio, Orientação a Aspectos, Modelagem de Processos de Negócios, AO-BPM 2.0, Business Process Model and Notation.

ABSTRACT

Businesses and systems are graphically represented to be effectively understood, and in their activities. Several languages and notations propose means for this representation, with its own syntactic, semantic and pragmatic characteristics. The Business Process Model and Notation allows the construction of more comprehensible and analyzable activity models for its stakeholders, without worrying about the interests spread by them, with the relevance of their activities or with the differentiation of the scope, thus representing monolithic blocks of information. Aspects Orientation sought to solve this question initially in programming, serving as a basis for the elaboration of languages for process-oriented systems and later for the Modeling of Business Processes. This work presents a comparison and evaluation of notations for business process modeling for aspect orientation modeling paradigm, in-depth analysis of one of them (AO-BPM 2.0) and its operationalization in a real scenario.

Keywords: Business Process Management, Aspect Orientation, Business Process Modeling, AO-BPM 2.0, Business Process Model and Notation.

Index

Tables Index

Figures Index

1 Introduction

Business process is a set of activities that produce value to a group of stakeholders. Business process models are abstract representations of such processes that could be made in different levels of abstraction. Process modeling is important in various stages of the business process management life cycle. (DUMAS, LA ROSA, *et al.*, 2013)

> There are many reasons for modeling a process. The first one is simply to understand the process and to share our understanding of the process with the people who are involved with the process in the daily basis. Indeed, process participants typically perform quite specialized activities in a process such that they are hardly confronted with the complexity of the whole process. Therefore, process modeling helps to better understanding the process and to identify and prevent issues. (DUMAS, LA ROSA, *et al.*, 2013)

Proportionally to the growth of the Business Process Management (BPM) field (MELCHER, 2012), we could notice the increasing complexity and content of business processes models (WEBER e REICHERT, 2008). At the same time, the aspect orientation paradigm from software engineering has emerged to decrease complexity (KICZALES *et al.*, 1997). Primarily poured to software development, aspects orientation brought and joined topics hitherto little or separately addressed in BPM. One of them was the support for the modeling of business processes focusing on interests, i.e. concerns, and more specifically on the crosscutting concerns (BRICHAU, CHITCHYAN, *et al.*, 2008), interlaced within the models.

Business process models have as well, as denoted in aspects orientation, different concerns in its composition (CAPPELLI, SANTORO, *et al.*, 2010). Until then no paradigm emphasized the crosscutting concepts, they used encapsulation or group of tasks, e.g. activity, to achieve a modularization and then a lower complexity. However, we find different syntactic elements spread, entangled and repeated within process models, such as: activities, connectors, gateways, events; increasing the complexity and the possibility of errors (CHARFI, HEIKO and MEZINI, 2010). These elements are not necessarily part of the core objective of the process or the business.

According to Cappelli et al. (2012), the main unit of modularity is the process itself, while a crosscutting concern is spanned in multiple processes. Typical examples of these elements in process models include logging, error handling, and security (an activity, which performs logging intertwines with other primary activities of a process). As the crosscutting behavior is scattered across some processes and tangled with other concerns, it is difficult to find and modify it. Therefore, process comprehension, evolution, and reuse are difficult to be practiced. Using aspect-oriented concepts, it is possible to modularize the crosscutting concerns and separate them from the main process flow.

Due to these issues, a notation to represent crosscutting concerns in process models was proposed previously by Charfi et al. (2010). that the intended proposal achieved goals related to conceptual and semantical perspectives, but the usage of the graphical notation showed up some drawbacks. After Charfi et al. other authors proposed their notations, each one improving the previous ones, like Cappelli et al. (2010), Jalali et al. (2012) and Tavares and Marinho (2014).

1.1 Aspect oriented approach examples

Many aspects oriented approaches are observed in informatics and business domains; An aspect oriented practice analogy in software could be the login web service provided by Facebook (FACEBOOK, 2016) for other applications; you do not have to sign any form or repeat information website by website, you just use your information hosted and provided by Facebook and enjoy its login service implemented in site using it. It is a tradeoff, Facebook deals with the information about its users and the website using its service deals with its core activities; e.g. Future Learn (FUTURELEARN, 2016) is a Massive Open Online Course (MOOC), with the mission: “FutureLearn’s mission is to pioneer the best social learning experiences for everyone, anywhere.”, the login is just an aspect to access its services. In that case, the services could be implemented as an aspect.

Picnik (PICNIK, 2013) was an online photo edition website. Bought by Google (RAO, 2010) to be a photo editor for his applications: “Since joining Google in 2010, the Picnik team has been working on Picnik while helping to create photo editing magic in Google's products. But now we get to focus.”. Google was not an editing images organization, instead of wasting resources “inventing the wheel” he just solved the problem of the “image edition” buying a famous photo editing solution, as a product, and adapting it for his needs.

Google applications handle their main activities, and another (in this case, Picnik) handles a soft-goal activity.

The mission and vision of a hospital does not involve (or should not) cleaning, security, mechanical or electrical engineering, software development, buying surgery material, etc. However, these activities are needed in a hospital, some of them to comply with laws or internal rules and others because they are helping the business to improve its value.

Printers often present problems and malfunctioning. In the hospital case, a doctor cannot write a prescription, but not fast, standardized and with equal quality as a printed one, so the printer become a problem. This equipment, also, have too many differences between models and fabricants (no established standard), so some printers from the same fabricant can be completely different from each other. The business process of a hospital need printers and need them functioning properly. Printers and printing are, in some way, directly involved in the mission, vision or values of this business? Directly not. The hospital can emit all his documents handwritten, but it will obviously hurt the efficiency of processes, negatively impacting the profit and income. An entrance form completely handwritten takes much more time than a printed one and is more, much more, prone to errors. A solution for this aspect, that indirectly impacts the hospital processes, is outsourcing the printer problem. Rent the printers and the built-in maintenance service, the hospital does not need to hire a printer technician just to do this, and so on the printers vary from each other, worsening the situation. The same with the cleaning, security, software development, etc. They are examples of aspects.

Smart Fit (CARVALHO, 2014) is a, initially, Brazilian network of academies who strictly provide bodybuilding, weight training and aerobics for a cheaper price them all its concurrency. They saw a lack of a specific service, all the gyms offered many services in addition to the first mentioned in this paragraph. So, Smart Fit provided strictly this service with a too much lower price than its competitors. The formula is simple: i) no dedicated workout or classes rooms; ii) automatized processes; and iii) a simplified service. Everything else is outsourced. While other gyms stayed with their old and outdated processes, Smart Fit stole the show and won a big slice of this market providing only part of the services the others provide, precisely the part that the target audience wanted the most, for a very lower price (OLIVEIRA, 2015) (CARVALHO, 2014). Smart Fit categorized the services that the other academies consider essential as aspects, carrying them sporadically and outsourced. This reduction in services allowed it to focus on what considers essential, offer better equipment and infrastructure and have high visibility in the market, among all social classes.

1.2 Objectives

Some notations have been proposed in the literature to represent aspects of business processes: AO4BPMN (CHARFI, HEIKO and MEZINI, 2010) and Aspect-Oriented Business Process Modeling Notation (AOBPMN) (JALALI, WOHED and OUYANG, 2012) as an extension of Business Process Model and Notation (BPMN); Aspect-Oriented BPM (AO-BPM) (CAPPELLI *et al.*, 2010) and Aspect-Oriented BPM 2.0 (AO-BPM 2.0) (TAVARES and MARINHO, 2014) as BPMN symmetric. The overall objective of this project is, respectively: to make a comparative analysis of these notations based on the following criteria: pointcut; join point; advice; aspect naming; precedence; encapsulation of aspects (aspects in aspects); modeling tool to represent and functionality in modeling tool(s) to represent and subsequently focus on improvement proposals only for the AO-BPM 2.0 through a case study, showing AO-BPM 2.0 advantages and disadvantages.

The research question that guided this study was enunciated as: To what extent are AO-BPM, AO-BPM2.0 and BPMN suitable for an abstraction (use, express and enable understanding) of aspects orientation in business process models built with their notations?

To accomplish these goals and answer the research question, the following steps were executed: (1) analysis and comparison of the aspect-oriented notations for modeling business processes; (2) further analysis and extension of AO-BPM 2.0; (3) case study with the symmetric notations to BPMN using the `LOC` metric to measure the results.

1.3 Structure

This undergraduate thesis is organized as follows: Chapter 2 provides a research background and theoretical basis of the relevant definitions and terms presented; Chapter 3 provides an analysis, review and comparison of four aspect-oriented business process modeling notations; Chapter 4 goes deeper in the AO-BPM 2.0 specification and suggesting improvements on it, this being the notation chosen for more detail in this work; Chapter 5 present the case study, BPMN, AO-BPM and AO-BPM 2.0 operationalization in a real setting; Chapter 6 discusses what has been built until this point at work; and Chapter 7 concludes the undergraduate thesis.

2 Research Background

This Chapter explains the concepts and definitions that provide the basis for the undergraduate thesis scope and to introduce the AO-BPM 2.0. Beginning with the Business Process Management (BPM) and following to Aspect Oriented Business Process Modeling (AO-BPM).

2.1 Business Process Management

A business process is a high-level component of a business that is comprised of several lower-level business activities; it delivers value to organizational stakeholders, and the business activity is a low-level component of a business that makes up a part of a business process; it consumes and drives costs (BYTHEWAY, 2014). Every business can be seen as a collection of business processes. Some of these processes are part of larger encompassing processes (LAUDON and LAUDON, 2013).

Plenty of material about the growth of interest in business processes by organizations in the last years are found, (WEBER and REICHERT, 2008) (BALDAM, VALLE and ROZENFELD, 2014) (RUSSEL and VAN DER AALST, 2016) (OLIVEIRA *et al.*, 2012) (FRANZ, KIRCHMER and ROSEMANN, 2012) (PAIM *et al.*, 2009) (SWENSON and VON ROSING, 2015) this behavior can be explained by Laudon and Laudon (2013): “By analyzing business processes, you can achieve a very clear understanding of how a business actually works. Moreover, by conducting a business process analysis, you will also begin to understand how to change the business to make it more efficient or effective.”

> Business process management (BPM) is a discipline involving any combination of modeling, automation, execution, control, measurement, and optimization of business activity flows in applicable combination to support enterprise goals, spanning organizational and system boundaries, and involving employees, customers, and partners within and beyond the enterprise boundaries. (SWENSON and VON ROSING, 2015)

As other key managerial variables like products and services, customers and employees, and physical or digital assets, the business processes need to be managed and BPM came as a discipline to cover that subject (BALDAM, VALLE and ROZENFELD, 2014). Business processes are susceptible to a series of variations, changing to best satisfy the customer requirements, to improve internal processes, adapt their products and services, business evolution, mergers and acquisitions, reuse of parts of the process to be discarded, inclusion of parts of others processes, coexistence of different versions of the same process, etc. (ROLLAND and NURCAN, 2010) and BPM handle, not limited on them, these topics.

The business processes of an organization are not only the ones that produce products, services and values or achieve organizational goals. There is a plethora of other activities that, if not performed, make impossible (or much harder) the achievement of the mission, vision or marks of the business, e.g. print a form, clean a meeting room, login in an information system, audit a ledger book, enable accounting transparency, maintain sustainability, transport packages, sign documents for authentication, etc. These are the kinds of activities the aspect oriented paradigm aims, and this work is dedicated to.

Business competencies are non-core commodities in an organization. Around 80% of the business processes in an organization are commodity processes that do not add to the differentiation or competitiveness of the organization, impacting in the process modeling (ROSING and KIRCHMER, 2015).

2.2 Business process modeling

Business process modeling, the activity of recording and representing the processes of an enterprise, is an important part of information modeling, which is the recording and depiction of the persistent and future arrangement of information assets of an organization in a structured or formal manner (SWENSON and VON ROSING, 2015).

To better understand the relevance of process modeling the list below enroll the benefits associate to this practice, not limited to them:

a) Provide a guided approach to gain understanding of the most important aspects of business process (SCHERUHN, VON ROSING and FALLON, 2015) (PAIM *et al.*, 2009) (RUSSEL and VAN DER AALST, 2016);
b) Business process models serves as a design blueprint for the subsequent software development and deployment activity (WESKE, 2012) (RUSSEL and VAN DER AALST, 2016);

c) Business process models provides an effective means of communicating their intention and operation in an unambiguous way to the various parties involved in their operation (PAIM *et al.*, 2009) (RUSSEL and VAN DER AALST, 2016);
d) Increasingly span organizational boundaries, and cross-organizational business process automation offers opportunities for enhancing and optimizing these processes in ways that were previously not possible (BYTHEWAY, 2014) (RUSSEL and VAN DER AALST, 2016);
e) Business process models provide a good starting point for all kinds of analyses (DUMAS *et al.*, 2013) (RUSSEL and VAN DER AALST, 2016);
f) Reflect compliance with laws and regulations, some of them requiring companies to identify the financial and operational risks inherent to their business processes and establish the appropriate controls to address them (RUSSEL and VAN DER AALST, 2016);
g) Enable business process transparency and visibility (FRANZ, KIRCHMER and ROSEMANN, 2012) (BALDAM, VALLE and ROZENFELD, 2014);
h) Identify and evaluate the value adding activities (SCHERUHN, VON ROSING and FALLON, 2015) (DUMAS *et al.*, 2013) (BALDAM, VALLE and ROZENFELD, 2014);
i) Define metrics for the process management (BALDAM, VALLE and ROZENFELD, 2014);
j) Train operators in the process (BALDAM, VALLE and ROZENFELD, 2014);
k) Compare business process with other through benchmarking (BALDAM, VALLE and ROZENFELD, 2014);
l) Plan human resources, helping and supporting the knowledge management (PAIM *et al.*, 2009) (BALDAM, VALLE and ROZENFELD, 2014);
m) Assist the commitments accomplishment in a consistent, predictable and reproducible way (PAIM *et al.*, 2009) (BALDAM, VALLE and ROZENFELD, 2014);
n) Create a culture and share a common vision through the organization using the same language in the operationalized models (PAIM *et al.*, 2009);

Modeling (the activity of “build” models) creates abstract representations, with greater or lesser degree of formality, of the reality in a given context. It is the most visible phase of BPM (BALDAM, VALLE and ROZENFELD, 2014). A model can be represented in many ways, from verbal description to graphical and technical notation exclusively created to

represent and show this kind of information. In this work, we use the combination of graphical and textual representations, the most common way to represent processes in organizations.

Business process models can be categorized in many forms. First, we separate them in three main groups: for software design groups, formal technical and dedicated for BPM techniques (RUSSEL e VAN DER AALST, 2016). Software design groups intend the development and enactment of Process Aware Information Systems (PAIS), Business Process Management Systems (BPMS), and other process based information systems. They are represented by Business Process Execution Language (BPEL), UML (Unified Modeling Language), YAWL (Yet Another Workflow Language), etc. Formal techniques formally represent processes pulling out the potential for ambiguities and inconsistencies, fully defined operational semantics ensuring these problems do not happen. They are represented by π-calculus, Petri Nets, YAWL, Linear Temporal Logic (LTL), etc. Dedicated BPM techniques, also known as conceptual models, specifically capture business processes; they are not primarily worried about formal semantics or implementation of a system, but to represent the business processes of an organization, helping the business to achieve their goals and values, the previous list presented in this Chapter is almost dedicated to this group of process models. They are represented by EPC (Event-driven Process Chain), UML activity diagrams, Workflow Nets, BPMN (Business Process Model and Notation), etc.

Descriptive models are used to share a common understanding of the business with the process stakeholders, analytic models with the purpose of improving a business process in order to increase efficiency of the overall organization and executable with the purpose of providing direct Information Technology (IT) support to a business process. The analysis in this work is guided by these three goals, describe, analyze and enact.

The notations studied are based in BPMN, currently one of the most widespread and operationalized business process technical notations (RUSSEL and VAN DER AALST, 2016) (DUMAS *et al.*, 2013) (BALDAM, VALLE and ROZENFELD, 2014) (SHARP e MC DERMOTT, 2008). BPMN is an informative, conceptual, flowchart, business process, swimlane modeling notation.

Business process models conceptually modeled for BPM have many data collected on them, normally activity, behavior, resource, relation between activities, agent, information, information entity, event, validation and modeling procedure (SHARP and MC DERMOTT, 2008). In the aspect oriented approach the important collection and representation are the aspects and soft-goals.

Business Process Modeling is a key element when aligning business processes with the requirements of an organization. With the right methodology and appropriate artifacts, it is possible to provide a clear, complete, accurate, and actionable framework for information and process modeling (SCHERUHN, VON ROSING and FALLON, 2015). However, overflow of information can lead to wrong decision making, damage to look for what you need it most and nonuse of relevant useful information (BALDAM, VALLE and ROZENFELD, 2014). The separation of interests and concerns in a process model, modularizing them, can unburden this excessive information.

The methods and methodologies available to perform the modeling task are associated, respectively, with the language that explicit the semantics and syntax for the symbolic representation of processes and the conceptual approaches that guides the activities needed to undertake the process modeling (PAIM *et al.*, 2009).

2.3 Modularization in Business Process Models

Modularity in Business Process Models is concentrated basically on levels of abstraction within process models, and for lower granularity levels, definition of its atomic elements, which can be reused along the diverse diagrams. Representing all functional and strategic purposes of an organization in a single process model is normally almost impossible, therefore processes are usually fragmented into a top-down or bottom-up approach.

In the top-down side and using the strategy of divide and conquer, processes are fragmented and reunited to represent the value chain. The Value Chain (BALDAM, VALLE and ROZENFELD, 2014) presents a high degree of abstraction, with the highest levels independent of the lower levels and without reciprocity.

As a mean to understand the chaining of processes and value aggregation generated by them with strategic or functional goals, the value chain is a very used tool (BALDAM, VALLE and ROZENFELD, 2014), created initially to manufacturing business but applicable to other types of business with a degree of adaptation. The value chain model identifies two types of business activity – primary and secondary (support) – and provides a framework for organizing the detail with them. The primary activities fulfill the value-adding role of a business and the secondary activities are required to control and develop the business over time, and to facilitate the working of the primary activities (BYTHEWAY, 2014).

The value chain is just a vehicle for exploring options and understanding consequences: it helps to shape a discussion rather than prescribe a solution and helps to get

beyond the detail of current arrangements in order to see the bigger picture in relation to the whole business (BYTHEWAY, 2014).

The bottom-up side allows for more experimentation and a better feeling for what is needed at the bottom (WIKIPEDIA, 2016). If top-down is divide and conquer, bottom-up is dynamic programming, the collection of smaller problems optimized and resolved can build, piece by piece, the solution for the whole (BELLMAN, 1954).

Sharp and McDermott (2008) defends the bottom-up strategy by classifying the top-down as “all-too-common phenomenon of the business process that looks suspiciously like an organizational unit”, so it’s more like a customary theoretical practice them a good practice at all. “Other than just being easier, three advantages of this approach are (a) the clients always give you more detail than you want anyway, (b) it’s easier to start by capturing detail than anything else, and (c) people like to see where the processes came from with respect to activities they recognize.”, but they defend the mixing of the strategies, depending on the stage of the business process lifecycle and so on.

The bottom-up can more proximately involve the stakeholders, collaboratively enhancing the process initiative (PRILLA e NOLTE, 2012). Gives the analyst a best comprehension of the punctual problems or specifics of the process, not available from a top-down, more generic, approach.

According to van der Aalst (2000): “when modeling complex processes, a hierarchical method of description is often an absolute necessity.” This modularization strategy, based on the recognition of sub-processes, has another important advantage, since it allows reusing the already defined processes. “If a particular sub process recurs several times, one definition used repeatedly will suffice. The reuse of (sub) processes often makes it possible to model a complex process more quickly".

Jablonski and Götz (2008) claim another common modularization issue in BPM; they state that users of process models need different types of visualization perspectives depending on their purposes. The authors identify five main perspectives for a basic process modeling notation: functional (process steps); data flow (data used in a process); operational (which operation is invoked to execute a process step); organizational (agents responsible to perform process steps); behavior (causal dependencies among elements).

A business process model acts as a blueprint for a set of business process instances, and each activity model act as a blueprint for a set of activity instances (WESKE, 2012), we can deal with business process models as information objects used by information systems (BYTHEWAY, 2014). Bytheway (2014) uses the analogy with housing in information systems

(describing the actors involved in the construction of information systems and the construction of a house), and Weske (2011) classifies the business process models as blueprints, the housing analogy for modularization in BPM are used so. When we see the blueprint of a house it shows the required information to build it, like the sections, elevations, width, height, plumbing, electrical network, ambient names, cut lines, and so on, but using a layer visualization we can analyze graphical data one by one, we can omit some of these items, but can we build a house without this information? That is modularization question. When we build hundreds of houses or apartments with some of this information equal, we do not need to build new blueprints, observing the context we can use the same. The same with information in business process models, we do not need to recreate a whole model or activity, and we reuse the already built. Of course, this analogy is not literally perfect, but can provide a pragmatic view of modularization in business process models.

Modularization mechanisms were already available in UML and EPC (LA ROSA *et al.*, 2011). Three techniques of modularization can be seen: horizontal, vertical and orthogonal.

The purpose of horizontal modularization is to increase maintainability by breaking down a process model into smaller and more easily manageable parts, the ownership of which may be assigned to different users. Hence, to facilitate collaboration. To reduce clutter in those models where long or crossing edges cannot be avoided. To foster reuse of modules within the same process model or across different process models (LA ROSA *et al.*, 2011).

The purpose of vertical modularization is increase understandability of large process models by “hiding” process details into sub-levels. To decrease redundancy and foster reuse by referring to a sub process from several places within the same process model or from different process models in a repository. The maintenance burden of a process model (repository) is also decreased, as a change to a sub process needs only be performed in one place (LA ROSA *et al.*, 2011).

The purpose of orthogonal modularization is to enable a separation of concerns and distribution of responsibilities. To facilitate maintenance of individual, concern-specific process models (LA ROSA *et al.*, 2011). Orthogonal modularization is the pure BPMN technique to handle crosscutting concerns.

This notion was necessary to a better understanding of modularization and encapsulation of the aspect oriented notations, because we find two types of notations representing aspects in business process models: those who hurt the semantics of BPMN and cannot be considered using that notation, even in a symmetrical solution, and those extending

it, don not hurting its semantic and respecting the official formalism and specifications (OMG, 2011).

We provide a comparison of three business process model representations: BPMN, Aspect Oriented Business Process Management (AO-BPM) and its notation, and Aspect Oriented Business Process Management 2.0 (AO-BPM2.0) and its notation. Each of them with their proper methodology and appropriate artifacts to guarantee the semantic and syntactic qualities to represent their process models, with the subsequent conceptual approaches and guides for process modeling.

2.4 Business Process Model and Notation

BPMN does not offer explicit syntactic or semantic to represent aspects, crosscutting concerns and soft-goals (CHARFI and MEZINI, 2004). But it offers ways to modularize the information in the model, under its own syntactical limitation.

BPMN does not address separated well-encapsulated items, the elements are scattered all over the models, poorly placed and containing entangled modeling constructs (e.g., activities and events) related to the core business process and modeling constructs that address other concerns, resulting in very complex and monolithic business processes, hampering understandability, maintainability and reuse (CHARFI, HEIKO and MEZINI, 2010).

A business process is modeled to exemplify the BPMN in Figure 1.

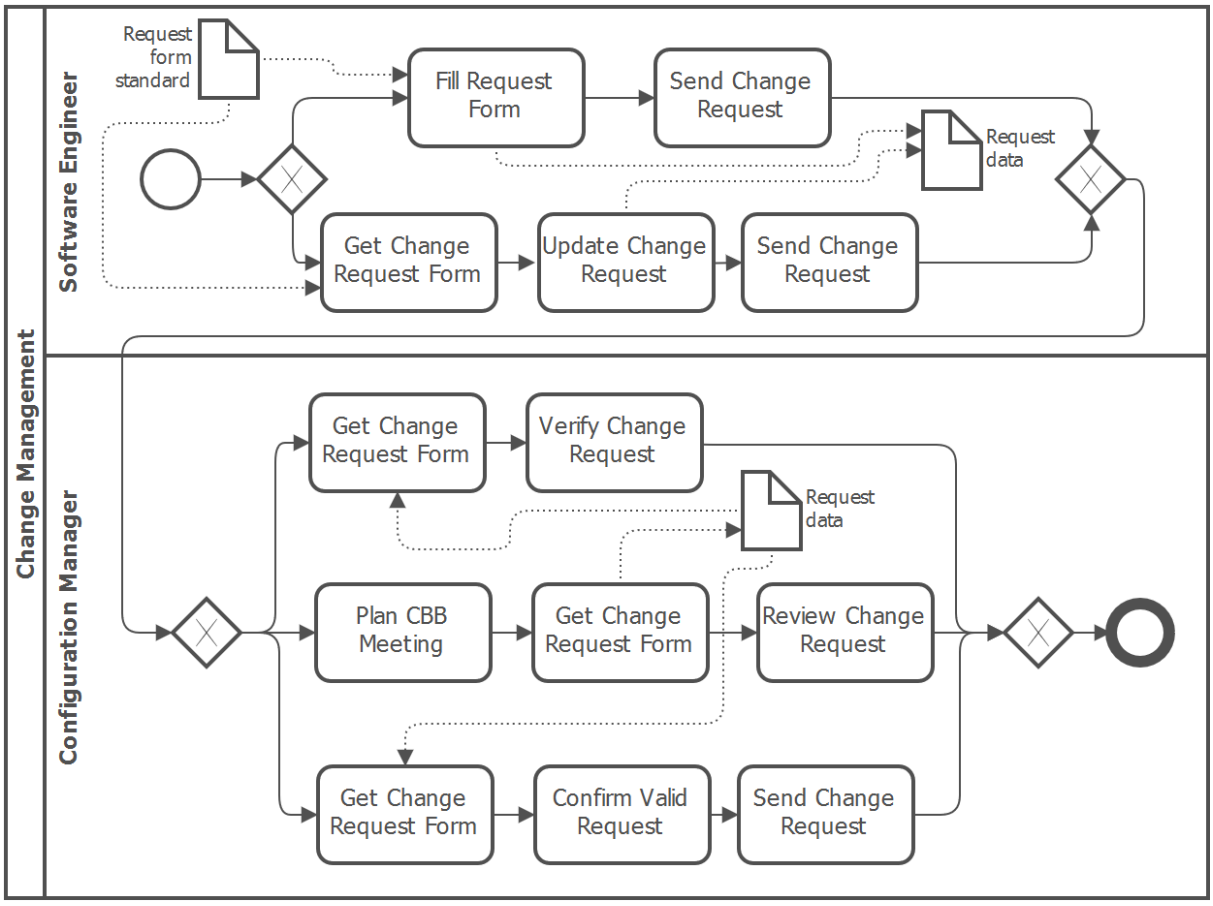

Figure 1: Change Management Process built using BPMN. Source: (CAPPELLI, SANTORO, *et al.*, 2010)

Using pure BPMN, we find one syntactical and one semantical way to modularize information in a model: using sub-processes or analyzing the activity semantical interpretation. The pragmatic modularization cannot be represented in a model.

2.4.1 BPMN modularization using sub-process

Modularization by sub-process is a syntactic form of this practice, content and meaning is not changed, it is only shown otherwise encapsulated. A Sub-Process is an Activity whose internal details have been modeled using Activities, Gateways, Events, and Sequence Flows. A Sub-Process is a graphical object within a Process, but it can also be “opened up” to show a lower-level Process (OMG, 2011). They are divided in five types: embedded, reusable, event, transaction and ad-hoc sub-processes.

Embedded sub-processes can be expressed as collapsed and expanded, all the other types have these two forms. The collapsed sub-process encapsulates its content from the current model, represented with a marker. The Sub-Process marker must be a small square with a plus sign (+) inside. The square must be positioned at the bottom center of the shape. A collapsed sub-process can be a single activity or other process; it is a frequently used

technique to modularize models with too much elements or too long. The expanded sub-process does not need the plus sign; it is a rounded rectangle with the sub-process itself fully represented in the same model. A collapsed sub-process example is presented in Figure 2, and an expanded one in Figure 3.

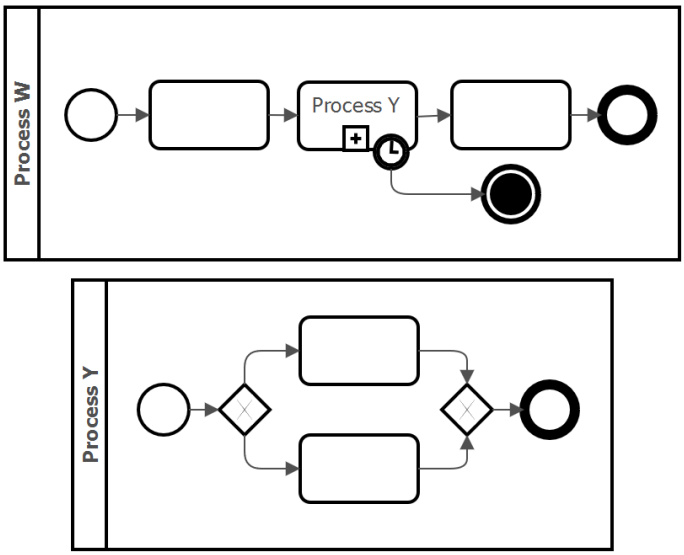

Figure 2: Example of collapsed sub-process.

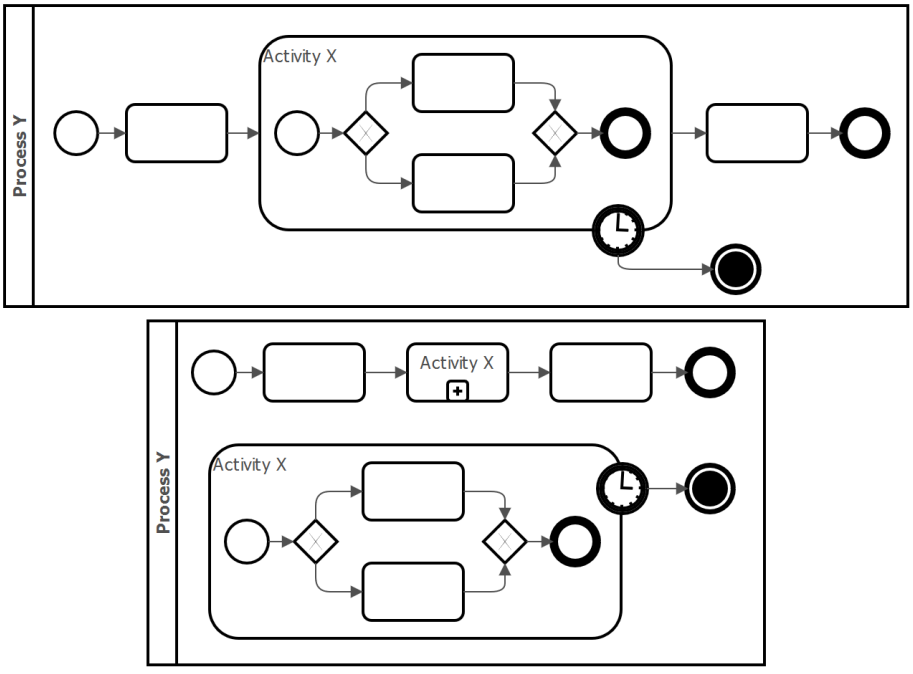

Figure 3: Example of expanded sub-process.

Reusable sub-process of BPMN 1.2 corresponds to the Call Activity that calls a pre-defined process. A boundary drawn with a thick line shall be reserved for Call Activity (sub-processes). It can call for global tasks or global processes, the elements determined this way are called Callable Elements. Callable Element is the abstract super class of all activities that

have been defined outside of a process but which can be called (or reused), by a Call Activity, from within a Process. An example of reusable sub-process is presented in Figure 4.

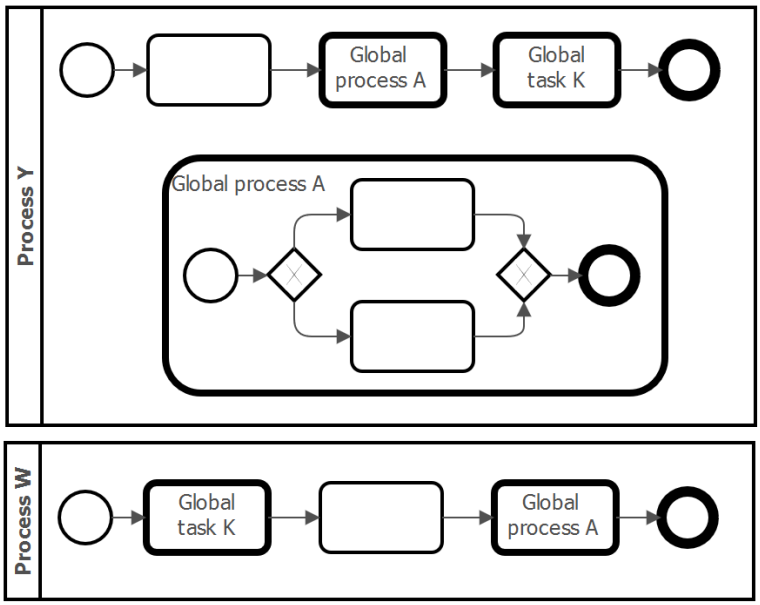

Figure 4: Reusable global activity and sub-process.

Event sub-process is a specialized sub-process that is used within a process (or sub-process). It is not part of the normal flow of its parent Process—there are no incoming or outgoing sequence flows. If a process is flowing, have an event sub-process attached to it and the condition defined in the initial event of this event sub-process is activated then two distinct ways are observed: if it is an interrupting event sub-process so the higher-level parent process stops and it only continues when the event sub-process is fully resolved; if it is a non-interrupting event sub-process so the higher-level parent process and the event sub-process continue concurrently. The process of Figure 5 represents an expanded event sub-process example, the message event with the dashed line is a non-interrupting event and the solid line is an interrupting event. An example of event sub-process, interrupting and non-interrupting, is presented in Figure 5.

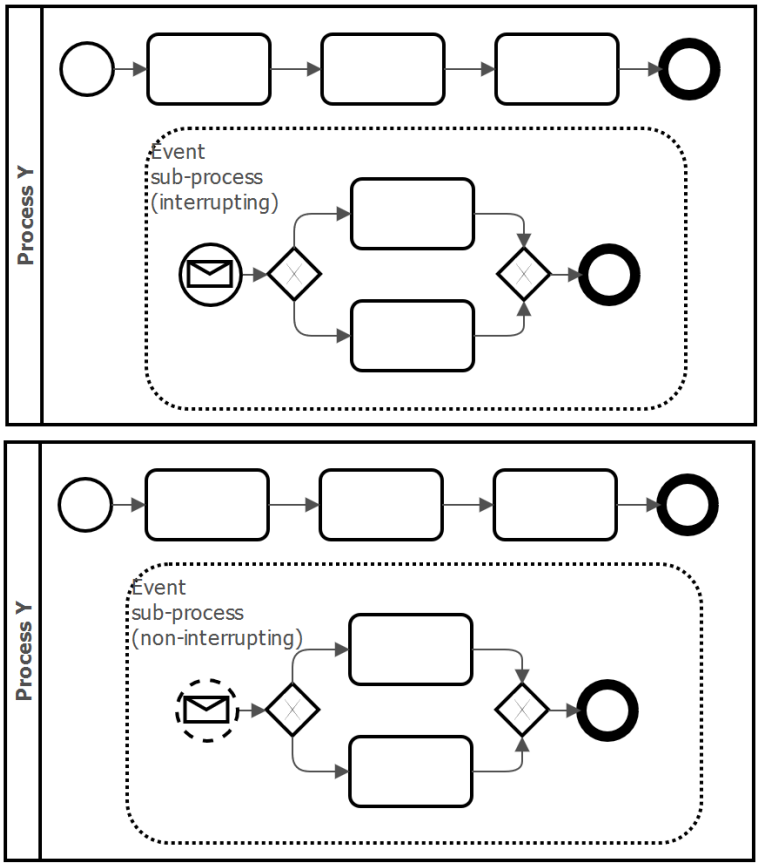

Figure 5: Event sub-processes, interrupting and non-interrupting.

Transaction sub-process is a specialized type of sub-process that will have a special behavior that is controlled through a transaction protocol (such as WS-Transaction). The boundary of the transaction sub-process will be double-lined to indicate that it is a transaction. They have three possible outcomes: successful completion, failed completion and hazard. The behavior at the end of a successful transaction sub-process is slightly different than that of a normal sub-process. When each path of the transaction sub-process reaches a non-cancel end event(s), the flow does not immediately move back up to the higher-level parent process, as does a normal sub-process. First, the transaction protocol needs to verify that all the participants have successfully completed their end of the transaction. Most of the time this will be true and the flow will then move up to the higher-level process. But it is possible that one of the participants can end up with a problem that causes a cancel or a hazard. In this case, the flow will then move to the appropriate intermediate event, even though it had apparently finished successfully. An example of transaction sub-process, collapsed and expanded, is presented in Figure 6.

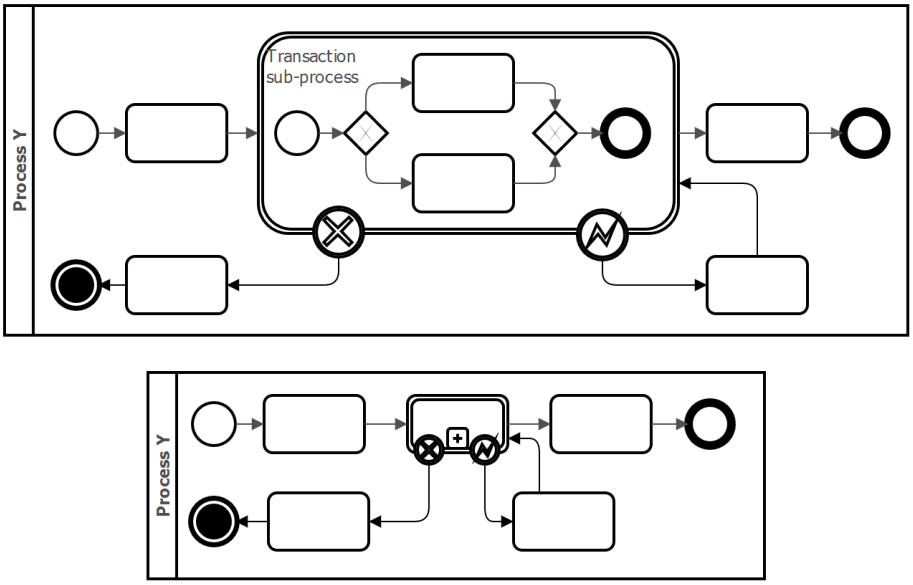

Figure 6: Transaction sub-process, expanded and collapsed.

Ad-Hoc sub-process is a specialized type of sub-process that is a group of activities that have no required sequence relationships. A set of activities can be defined for the process, but the sequence and number of performances for the activities is determined by the performers of the activities. A sub-process is marked as being ad-hoc with a “tilde” symbol placed at the bottom center of the sub-process shape. It is a challenge for a BPM engine to monitor the status of Ad-Hoc sub-processes, usually this kind of processes are handled through groupware applications (such as e-mail), but BPMN allows modeling of processes that are not necessarily executable, although there are some process engines that can follow an Ad-Hoc sub-process. The Ad-Hoc sub-process conception is an attempt to represent declarative or loose processes using BPMN. An example of Ad-Hoc sub-process is presented in Figure 7.

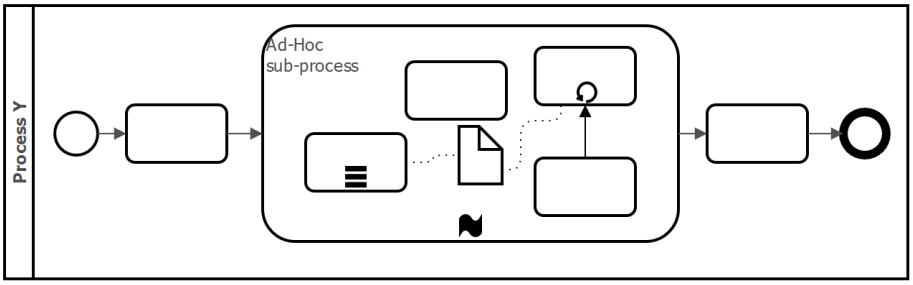

Figure 7: Ad-Hoc sub-process, expanded.

Using the syntactic modularization in Figure 1, the resulting model will be Figure 8.

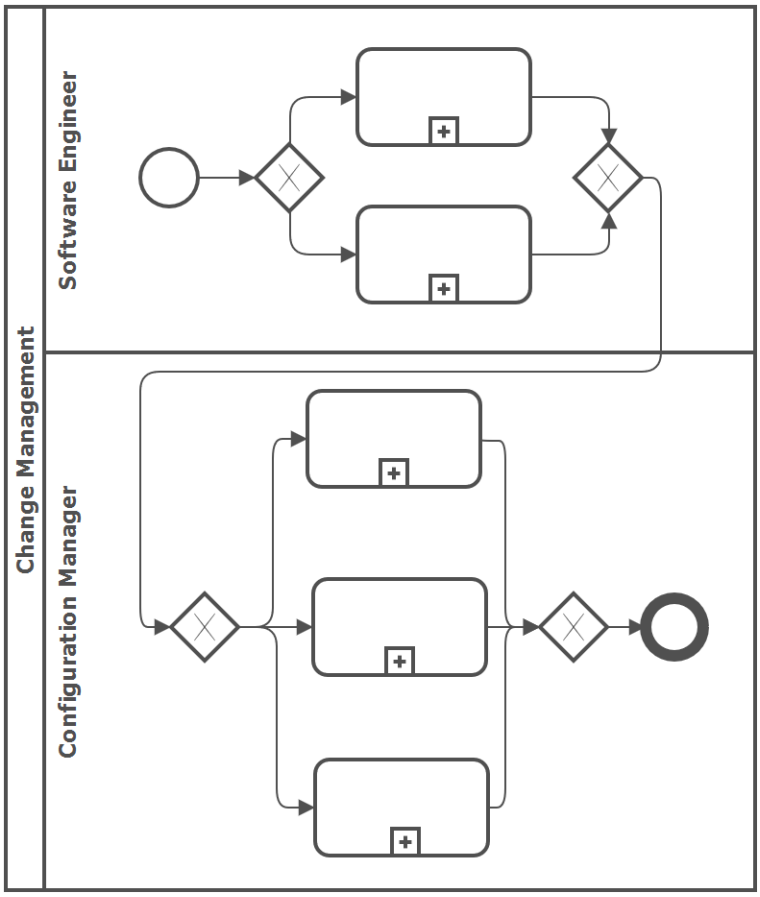

Figure 8: Change Management Process syntactically modularized.

Figure 8 represents the low-level son process of the process represented in Figure 1. So, it was a level two in the direct hierarchy of process in this scope. But we can modularize once more, resulting in Figure 9.

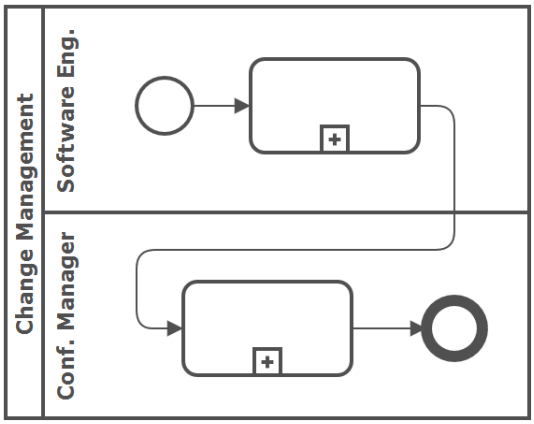

Figure 9: Change Management Process syntactically modularized in two levels.

That is the most exhaustively use of modularization using pure BPMN in the process scope represented in Figure 1. The elements in Figure 1 are contained in Figure 8 which in turn is contained in Figure 9.

2.4.2 BPMN modularization using semantic interpretation

An activity is a process step that can be atomic (tasks) or decomposable (sub-processes) and is executed by either a system (automated) or humans (manual). All activities share common attributes and behavior such as states and state transitions. An activity, regardless of type, has lifecycle generally characterizing its operational semantics.

The activities presented in Figure 1 were just represented and abstracted this way because a business process analyst or business expert defined as so. There are no impediments in BPMN semantics capable of directly influence the interpretation of the tasks or activities determined by the actors who designed them. So, a process modeler can just merge activities and tasks considering his understanding and expertise on the information to be modeled itself.

Figure 10 shows an example of the process in Figure 1 in a different interpretation, using the pure BPMN and modifying the semantic scope, different and subsequent activities are merged, without influencing the flow, e.g. Get Change Request Form merged with Verify Change Request.

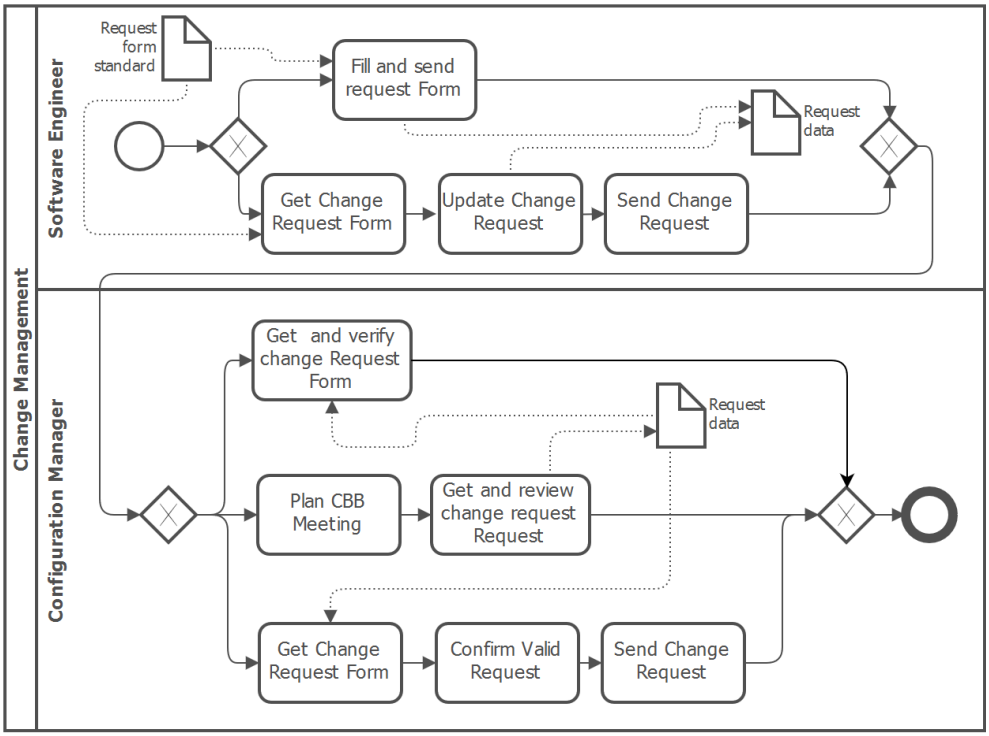

Figure 10: Change Management Process semantically modularized.

Functions are composed of processes, which are composed of activities, which are composed of tasks (SHARP and MC DERMOTT, 2008). Nevertheless, activities are composed of processes. The granularity of the tasks or activities in a model need to be taken into account (SHARP and MC DERMOTT, 2008), if they are so specific, granulated and atomic so two results can be seen: i) no one will ever execute it in the modeled way; ii) when

analyzing or monitoring it will always look wrong and full of errors, because no one is literally following religiously the process as the model says. In the other side, if they are so generic, not granulated and loose them everything can be done and anything can be measured, analyzed or monitored, causing all the possible scenarios validation, not just the right ones. A middle term is needed, and in this middle term the semantic modularization is observed.

2.5 Aspect-Oriented BPM

The goal of aspect-oriented software development is to support the programmer in cleanly separating components and aspects from each other, by providing mechanisms that make it possible to abstract and compose them to produce the overall system (KICZALES *et al.*, 1997). In object-oriented software development, we achieve the separation of interests by decomposing the application into individual objects (BRICHAU, CHITCHYAN, *et al.*, 2008). Other concerns remain fragmented by the code, remaining redundant. The aspect orientation decomposes the application in interests, modularizing them, reducing redundant excess.

An interest is a concern regarding the development of the system, its operation or another issue that is critical or somewhat important to any interested party (VAN DE BERG, CONEJERO and CHITCHYAN, 2005). The concerns may cross or be entangled in the system, or code. Some examples are: login, security policies, data storage, audit, prevention against bugs, regulation, supervision (KICZALES *et al.*, 1997) (BRICHAU *et al.*, 2008) (CAPPELLI *et al.*, 2010); they are classified as non-functional requirements, or soft-goals, regarding the core application objective in question (CAPPELLI *et al.*, 2010) named as aspects.

Some recent approaches propose also to modularize the crosscutting concerns within process models (CAPPELLI *et al.*, 2010) (TAVARES and MARINHO, 2014)[CITATION Ami11 \l 1046] (CHARFI, HEIKO and MEZINI, 2010). The modularization provided by the aspect orientation, through crosscutting concerns, is an application of the “divide and conquer” strategy argued by van der Aalst and van Hee (2000). Interests (concerns) permeating the processes of the value chain and crossing each other could be defined as aspects. Then, not only the sub processes and the activities could be modularized, but also other elements of the process, such as data, rules and resources.

The current scenario of business process modeling operationalizes modularization as sub processes and subjective activities (VAN DER AALST and VAN HEE, 2000) (SANTOS *et al.*, 2012) (JALALI, 2011). The modularization with the notation revisited benefits the size

and simplicity of the macro process (VANDERFEESTEN *et al.*, 2007) (MENDLING and REIJERS, 2008), the comprehension factor is based on experience, training and taste of the reader (RUSSEL and VAN DER AALST, 2016).

Using the analogy of the housing presented in the last paragraph of Chapter 2.3, the central air conditioning, the network plan, the colors scheme, the architectural adornments, etc. are examples of aspects in a house or apartment blueprint.

2.6 Aspect-oriented glossary

Aspect orientation paradigm, in software or process modeling, has its own terms to represent their concepts. These expressions are not assignable to other notations or paradigms, so they need to be clarified to better understanding of this work. They are presented below, sometimes adapted for process modeling context.

Aspect: Aspects are elements that modularize a certain crosscutting concern. They consist of one or more pointcuts and associated advices. In addition, they may define their own state with data objects (CAPPELLI, 2009) (CHARFI, HEIKO and MEZINI, 2010). An aspect is a cross-cutting module which consists of one or several pointcuts which relates to one or several advices (JALALI, 2011). Charfi *et al.* (2010) proposed a “light” and “heavy” syntax to AO4BPMN aspects, the definition of aspect remained, but the syntax changed.

Join Point: Join points are points in the business process model where modeling elements implementing a crosscutting concern can be integrated (CAPPELLI, 2009) (CHARFI, HEIKO e MEZINI, 2010). A join point can be related to one or several advice(s) via certain rules. Such a rule is known as a pointcut, a join point to which an advice is linked via a pointcut is called an advised join point (JALALI, OUYANG, *et al.*, 2015).

Pointcut: A pointcut is a construct that allows the selection of related join points (CAPPELLI, 2009) (CHARFI, HEIKO and MEZINI, 2010). A pointcut is a construct that selects join points by matching certain characteristics and collects context at those points (VAN DE BERG, CONEJERO and CHITCHYAN, 2005).

Aspect Precedence: Aspect Precedence controls the advice execution order, which is the order in which the advice is applied, in cases that advice in more than one aspect applies to a join point (VAN DE BERG, CONEJERO e CHITCHYAN, 2005). Cappelli (2009) presented the precedence in AO-BPM using a “pointcut language”, dedicated to it. Jalali et al. (2012) elaborated a consistent precedence method in (JALALI, WOHED and OUYANG, 2012), operationalizing the precedence in the main process with an intermediate event object

and textual annotation artifact. Tavares and Marinho (2014) use contoured elements directly in the main process, following the flow, the precedence is the order of these elements.

Advice: An Advice is a method-like construct that provides a way to express crosscutting action at the join points that are captured by a pointcut. There are three kinds of Advice: Before advice, After advice, Around advice (VAN DE BERG, CONEJERO and CHITCHYAN, 2005). A process, which models a cross-cutting concern, is called an advice. The advices representing similar concerns can be classified into the same group, which is called an aspect, e.g. Security or Logging Aspect (CAPPELLI, 2009) (JALALI, WOHED and OUYANG, 2012).

Concern: A Concern is a specific need that must be addressed in order to satisfy the overall system or process goal. There could be two types of concerns: Core concern and Crosscutting concern (CAPPELLI, 2009) (VAN DE BERG, CONEJERO e CHITCHYAN, 2005).

Core Concern: A Core Concern is a concern that captures the central functionality of a module in a system or process (VAN DE BERG, CONEJERO and CHITCHYAN, 2005).

Crosscutting: Crosscutting is the scattering and tangling of concerns arising due to poor support for their modularization (VAN DE BERG, CONEJERO and CHITCHYAN, 2005), e.g. the BPMN aspect modeling limitation (CHARFI, HEIKO and MEZINI, 2010) caused by the monolitical representation of information. We distinguish two levels of crosscutting: Crosscutting at design level is the scattering and tangling of concerns at modeling level and; Crosscutting at implementation level is the scattering and tangling of code which belongs to different concerns (VAN DE BERG, CONEJERO and CHITCHYAN, 2005).

Crosscutting concern: A Crosscutting Concern is a concern that captures requirements that cross multiple modules or processes in a system, single process or process portfolio (CAPPELLI, 2009) (VAN DE BERG, CONEJERO and CHITCHYAN, 2005).

Weaving: One process represents the core-concern, while others represent cross-cutting concerns. Therefore, a mechanism should be defined to join these concerns with each other. This mechanism is called weaving service (CAPPELLI, 2009). Weaving service could perform at the design-time and at the run-time. The run-time weaving service add more value to the aspect oriented business process, because it allows more flexibility in defining and assigning concerns to a core-concern. The weaving service should define a mechanism to apply the appropriate ordering between different concerns (JALALI, 2011).

3 Analysis of aspect oriented business process modeling approaches

This Chapter introduces the aspect-oriented approaches used in this undergraduate thesis, AO-BPM and AO-BPM2.0, clarifying some concepts about the aspect-oriented paradigm, focusing on the modeling and representation of aspects in business process models.

BPMN, now at version 2.0.2, evolved from version 1.0 trying to solve some problems and missing gaps. The first version of BPMN had the initial focus to create a business process modeling language for business people that could also be executed by available BPMSs (ROSING, WHITE, *et al.*, 2015). Even if it is the most used notation it does not provide syntactical, semantical and pragmatic modeling signs and content to many modeling goals and strategies, like goal-oriented, value-oriented, sustainability-oriented, case management, business rules, data, change management, decision, etc. modeling purposes; not that BPMN cannot represent these matters, but the primarily goal of the BPMN notation use, with their objects, artifacts, documents, semantics, etc., is not represent those purposes (OMG, 2011). With the need to represent, these items initiatives created languages and notations to express them with formal syntax and semantics, e.g. Change Management Modeling Notation (CMMN) and Decision Modeling Notation (DMN). BPMN v1.0 was released in 2004, may, and the v1.1 in 2008, January, and v1.2 in 2009, January; v2.0 in 2011, January. Since them the notation received only one minor alteration, in 2014, January, versioned 2.0.2[CITATION OMG11 \l 1046].

CMMN v1.0 was released in 2014, May; and the DMN was released in 2015, September, as v1.0 and v1.1 subsequently came in 2016, June.

DMN was created to complement the BPMN standard. Together, BPMN and DMN may be used to build process models more efficiently (DEBEVOISE e TAYLOR, CreateSpace). However, the DMN do not include syntax or semantics directly into BPMN,

DMN was designed to complement BPMN and provides a separation of concerns between the decision and the process.

Case management is a hot topic in BPM (ROSING *et al.*, 2015), but CMMN do not formally and directly addressed and represented in the BPMN (OMG, 2011). Some authors believe that the best course is that the Object Management Group (OMG) consolidates the two specifications in the next update to BPMN, v3.0 (ROSING *et al.*, 2015).

The aspect-oriented modeling, not in the above list, is one more example of non-contemplated conception by BPMN in its purposes. Differently from CMMN and DMN, the aspect-oriented approach is not supported by an initiative like OMG, but by independent groups around the world, incrementing and researching the topic.

Each concept listed emphasizes a business process characteristic, which can be change, decision, sustainability, goals, values, business rules, etc., in this concept we deal with aspects. So far, the aspect-oriented approach in business processes addresses the modularization as it main instrument (KICZALES *et al.*, 1997) (SANTOS *et al.*, 2012) (JALALI, 2011) (VAN DE BERG, CONEJERO and CHITCHYAN, 2005) (CAPPELLI *et al.*, 2010). Cappelli (2009), Charfi *et al.* (2010), Jalali (2012) and Tavares and Marinho (2014) proposed aspect oriented modeling notations, extending BPMN or symmetric to it, to represent conceptually aspects in business process models. The last of them was Tavares and Marinho, analyzing AO-BPM and releasing the AO-BPM 2.0 notation, with their improvements to AO-BPM notation. This work presents a briefly explanation and exemplification of these notations, and analyze in details only AO-BPM 2.0, this decision is explained in Chapter 3.8.

Aspect orientation meets business process management initially in the enactment phase, focused in system development with Business Process Execution Language (BPEL), since the aspect paradigm initiated in programming and not in conceptual modeling. The success of the Aspect Orientation for Business Process Execution Language (AO4BPEL) inspired researchers to develop techniques to support aspect orientation in other BPM topics as well, as in this work the Business Process Modelling.

Definition of aspects, crosscutting concerns, interests or soft-goals depends on the organization, it is not an information we can infer just by modeling. The analysis of value adding activities and data can help an analyst to elicit and identify properly this information from the business context. It is not trivial to identify or define crosscutting concerns (CAPPELLI, SANTORO, *et al.*, 2010), given the process interpretation and its subjectivity. Cappelli (2009) suggests a list of concepts that potentially can be aspects:

- An existing or new activity or an event can affect several parts of the process, including connectors.
- Activities appearing in several processes related to the same macro-process.
- Data replicated in the same process or in different processes related to the same macro-process.
- Events replicated in the same process or different processes related to the same macro-process.
- Common goals among different processes related to the same macro-process.
- A role, which performs activities in different processes, related to the same macro-process.

Tavares and Marinho (2014) adapted a methodology created by Sharp and McDermott (2008) to elicit aspects and soft-goals among the business process models. The adapted methodology consists in six steps listed and briefly explained here:

a) **Organizational culture evaluation.** The first step consists in discovering the organization mission, strategy and goals. Not limited only to them, it can be extended to encompass the vision, values, critical success factors, differentiation, etc.

b) **Elaborate a process map.** Observe the process in a macro scenario, the bigger picture. Enabling the visualization of transversal interests, crosscutting concerns, an information crossing and involving many functional areas, etc.

c) **Establish the scope of the processes.** Tavares and Marinho present a form to be filled, registering relevant information about the process itself, detailing it.

d) **Analysis of the stakeholders.** Stakeholders' evaluation, contextualizing each one involved with the process.

e) **Process enablers' investigation.** There are six enablers: workflow design, information systems, motivation and measurement, human resources, policies and rules and facilities. An enabler is a factor that can be adjusted to impact process performance.

f) **Develop a glossary of terms.** The heart of the data model, and the starting point for building one, is an agreed-upon list of core things or nouns, which are the entities. That is the glossary.

The proposals mentioned so far are manual, the aspects are subjectively analyzed by experts and then categorized as such, stemmed from use cases, event logs, requirements in natural language and natural language description (BRANDÃO, 2015). However, aspects can

also be automatically detected using mining techniques. In AOP three approaches are used to find aspects in code: Clone, Cluster and Fan-in (BRANDÃO, SANTORO and AZEVEDO, 2015): "The Clone approach aims to discover the methods in the code that can be clones of others, i.e., duplicated code. The Cluster approach aims to group patterns of methods by their execution. For example, if the methods A, B and C are always performed together, they are grouped as an aspect candidate. Finally, the Fan-in approach calculates the fan-in measure of methods to identify aspects, i.e., the approach counts the number of times the method is invoked by other methods. And when a method is invoked more times than a defined threshold, it is identified as an aspect candidate."

Aspect identification is similar in automatic and human identification, based on the scenario, details and level of abstraction. The method evaluation is qualitative, depending on specialists' opinion to analyze if the aspects found were accurate (BRANDÃO, 2015). The automatic techniques can be used to assist and facilitate aspects identification, at least (BRANDÃO, SANTORO and AZEVEDO, 2015).

3.1 Aspect Orientation for Business Process Model and Notation (AO4BPMN)

Charfi *et al.* (2010) proposed the first relevant aspect oriented notation to include aspects in conceptual business process models. The authors defined join points, pointcut, advices, proceed nodes and aspects; but they do not clearly define the joint point and pointcut representation in the modeling notation, we cannot observe where aspects or encapsulated items are connected in the main flow in a model using AO4BPMN. It is important to observe that AO4BPMN does not hurt the BPMN syntactical or semantical rules, therefore being an extension to it, adding objects, not a symmetric solution, modifying them.

Charfi *et al.* (2010) did not build only the AO4BPMN notation but also the AO4BPEL, then the first resembles a software design phase object, slightly influenced by the latter.

Given the representative limitation of the proposed notation, AO4BPMN is not used here (representing Figure 1 with it), already criticized and analyzed in the next works presented here. Even in his complementary AO4BPMN work (WITTERBORG *et al.*, 2014) they did not explain or show the representation of joint points or pointcut in the models, as he says: “However, that extension had some open issues especially with respect to the lack of a concrete pointcut language and the lack of a weaving mechanism.”.

3.2 Aspect Orientation – Business Process Management (AO-BPM)

Cappelli (2009) proposed the AO-BPM Notation, which adopts a symmetric strategy for modeling aspects, using the same abstraction to represent elements of the notation. The crosscutting concern, a potential aspect, is represented in a specific swim lane and each swim lane represents a different crosscutting concern and could be described by a set of business process model elements. See example in Figure 11.

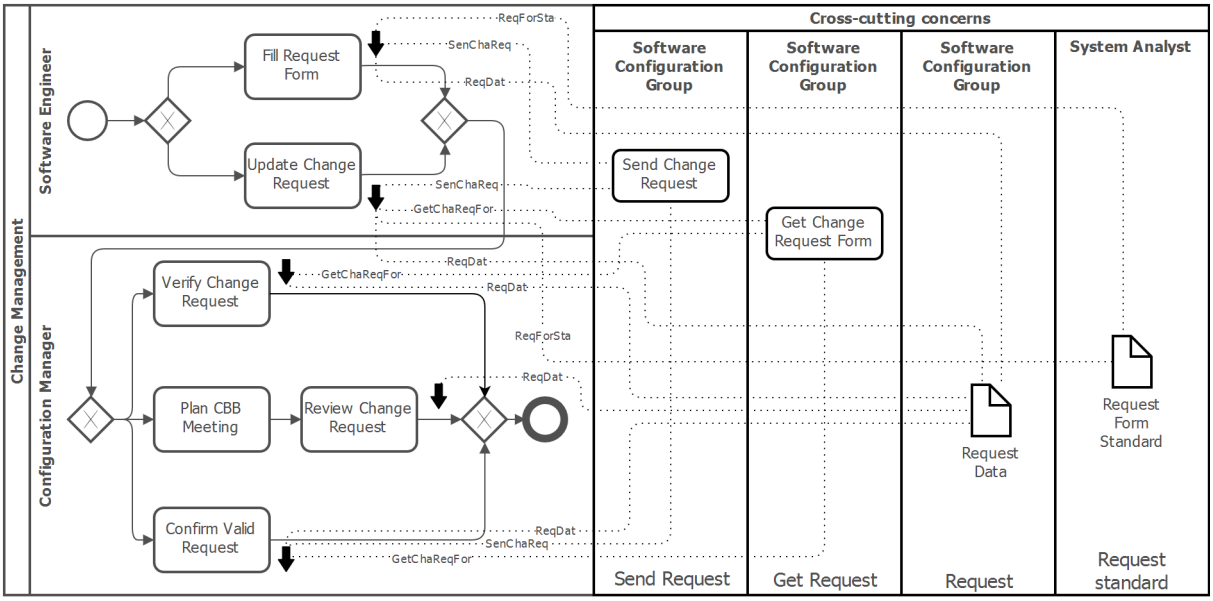

Figure 11: Change Management Process built using AO-BPM notation. Source: (CAPPELLI, SANTORO, *et al.*, 2010)

The process model is composed of several elements, which may be the target of crosscutting concerns. A crosscutting element affects other process model elements through the crosscutting relationship. The current elements of the process model are the potential join points, with no need to include additional information. Graphically, the join points can be represented by a ground element which locates near it, allowing that the source of the crosscutting relationships to be the crosscutting element, and the target be the ground element representing the join point. Figure 11 illustrates the join points with a black ground element. “Send Change Request” is a crosscutting concern, which affects the “Update Change Request”, “Fill Request Form” and “Confirm Valid Request” activities. “SenChaReq” is the label pertaining to the crosscutting relationship between them.

AO-BPM detailed the pointcut notation, defining patterns to write pointcut expressions. The pointcut notation represents, textually, the points where the aspect acts, and the moment this is being applied (before, after, during) at the core description. Therefore, the pointcut notation should allow for the combination of join points which express the inclusion

of crosscutting concerns in a process. Include primitive is the main clause of the pointcut notation, used in the advice part to specify the insertion of a crosscutting concern in a core process.

The join point in the AO-BPM notation is not represented in the model, making necessary a language to explicitly detail it, describe the advices and aspects presented and notify if they are “before”, “after” or “during” the elements, named pointcut language, represented in Figure 12.

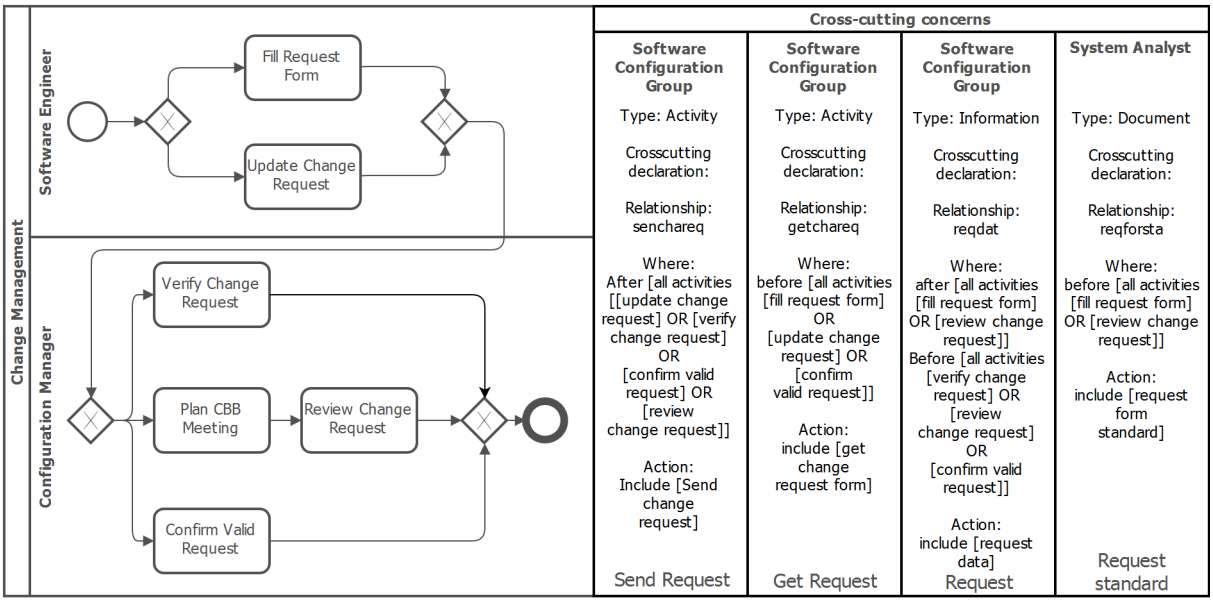

Figure 12: Change Management Process Pointcut Language. Source: (CAPPELLI, SANTORO, *et al.*, 2010)

One of the most important contributions of Cappelli (2009) work was a heuristic to identify aspects shown in Chapter 3. Among other contributions was a solid and structured analysis of aspect orientation not straightly focused in enactment and execution of process models but in conceptual representation of it, focused in a semantically better information representation.

Although, some AO-BPM drawbacks are discussed as follows:

- While the number of aspects and the size/complexity of process models increases, the representation becomes increasingly difficult to be learned;
- It does not consider the hierarchy within an aspect itself, aspects inside aspects and their genealogical level;
- The separated join point information disrupts the understandability; it is necessary more than a model to analyze the flow, representing the pointcut language;

- It does not consider, in the work presenting the notation, other elements than activities and data objects as encapsulated aspects, crosscutting concerns or soft-goals. It does not cover resources, other aspects or flows as aspects;
- The increased number of aspects might generate increased number of flows and number of relationships for them (Figure 11), hurting the model understandability and readability (TAVARES e MARINHO, 2014). Crossing edges need to be avoided, this concrete syntax behavior hurt the process model comprehensibility. In matter of scalability AO-BPM can be so polluted that end up reaching the incomprehensibility;
- The use of excessive text and data to represent the concepts can lead to error. In Cappelli *et al.* (2010), Figure 5 has an error in the relationship of the “Request Form Standard” document lane, instead of “ReqForSta” (noted at Figure 4 at same work) the relationship is “ReqDat”. This type of error can happen when working with too many different and compared artifacts.
- It does not use Advicesadvices names to identify and clarify about the aspect itself, if it is an accounting, billing, transport, login, etc. aspect. Neither in the main flow, with the crosscutting concerns pool, neither in the pointcut language.
- There is no modeling tool to model AO-BPM.

In the view of all those issues, we use here a new version of the AO-BPM, the AO-BPM 2.0 (Chapter 3.4), with extension to the original notation and the graphic elements reviewed.

3.3 Aspect Oriented Business Process Model and Notation (AOBPMN)

Jalali *et al.* (2012) proposed the AOBPMN notation, analyzing the AO-BPMN and AO4BPMN, trying to improve them and fill empty gaps. He not only proposed his notation, but also improved the AO4BPMN (WITTERBORG *et al.*, 2014) and was thanked by Charfi.

AOBPMN, as AO4BPMN, is an extension of BPMN, does not hurting its syntactic and semantic properties. Differently from AO-BPM, the encapsulated aspects do not connect themselves to the main flow by dotted arrows, but by intermediate conditional event marker attached to the activity, representing the pointcut. The joint point is present in the main model as an annotation attached to the intermediate conditional marker attached to the activity, stating if the encapsulated data is “before”, “after” or “around” (“during” in AO-BPM) the respectively pointcut. In this join point the order of the concerns called by the pointcut is described, e.g. if there are three aspects called in sequence, they will be ordered, by the

correctly flow order, by 1, 2, 3. The join points in AO-BPM are not graphically represented in the main model, but in the pointcut language.

The notation uses the name in the aspect, initiated in AO4BPMN, determining the modeled and represented concern, e.g. login, transparency, security, etc. It can be modeled in any tool or application enabled to model BPMN, since it is an extension of it. AOBPMN concepts, syntax and semantics can be adapted to other modeling languages and notations, as UML and EPC (JALALI, WOHED and OUYANG, 2012).

The intermediate conditional event attached to the activity leads the main flow to the encapsulated activities or tasks, executing them in the order presented by the join point and returning to where it was separated before.

Some AOBPMN drawbacks are discussed as follows:

- While the number of aspects and the size/complexity of process models increases, the representation becomes more and more difficult to be learned;
- It does not consider the hierarchy within an aspect itself, aspects inside aspects and their genealogical level;
- It does not consider, in the work presenting the notation, other elements than activities as encapsulated aspects, crosscutting concerns or soft-goals. Do not cover resources, data objects, other aspects or flows as aspects;
- Increasing the number of aspects might increase the number of annotations, and the number of text and space to place them trying to maintain the readability, hurting the model understandability and possibly the readability;
- The use of excessive text and data to represent the concepts can lead to error. The “tags” representing the pointcut can induce the modeler to errors;
- Advices use of software terminology, the first two brackets in the pointcut holds the condition if a gateway or decision was involved, but inside the encapsulated aspect and not explicitly in the main flow. That is not an easy concept to a business analyst understand.

Figure 13 shows the process of Figure 1 modeled using AOBPMN.

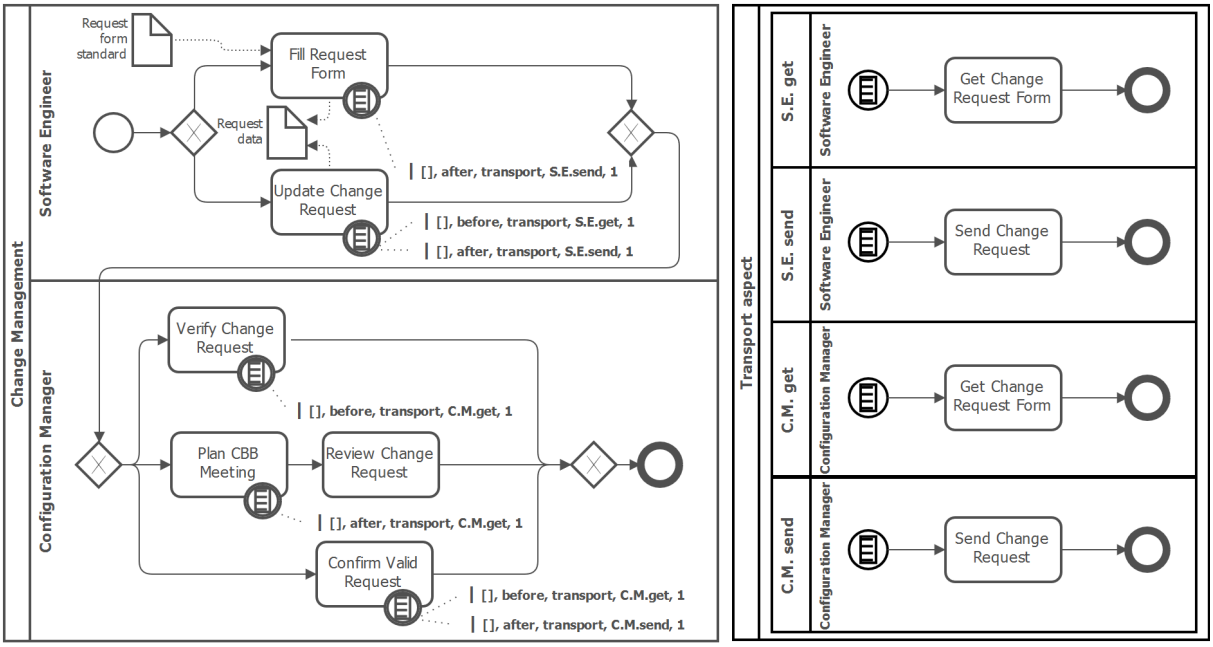

Figure 13: Change Management Process built using AOBPMN.

3.4 Aspect Orientation – Business Process Management 2.0 (AO-BPM 2.0)

Tavares and Martins (2014) had first proposed adaptations in AO-BPM (which is described in Chapter 3.1) to make it more suitable for understanding purposes. AO-BPM 2.0 adapts its predecessor removing the relational aspect connectors, crosscutting relationships, and inserting in its place numbers identifying each aspect. The AO-BPM 2.0 is still symmetrical to the BPMN since it does not propose changes in its syntax or usage.

The aspects are operationalized by AO-BPM 2.0 as follows:

a) Each aspect has its own pool, following the pattern shown in items (ii), (iii) and (iv);
b) A lane should be created for each aspect; it should contain an identification device used to trace the aspect in the business process model;
c) The actors must be specified, through "sub lanes", because the aspect will involve the actors that are defined in the original model. Thus, there could exist resources in the pool of the aspects that are not present in the main model;
d) The aspect is a "part" of the process, so it should have its dependence and its output in the business process model; however, it is not necessary to maintain the indicative flow on them. These flows should be implicit, beginning at the first activity and ending at the last that are in the pool aspect;

e) The pool representing the aspect may contain "sub lanes" that determine the actors involved in that particular aspect, connecting the elements to the resources involved in the aspect.

Concerning the diagrammatic issue, a number associated to the aspect corresponds to the join points, inside a circle with running trace of contour. There may be several join points in a single flow, and as so, in order to identify what is the execution sequence, it is used the direction of flow: the nearest junction point of the direction of flow indication will be the last executed and the most distant of this statement the first. Aspects representing documents, whether incoming or out coming, differ from others by the dashed contour.

Some AO-BPM 2.0 drawbacks are discussed as follows:

- It shows the encapsulation of activities and data objects, aspects may be composed of other aspects, recursively and with hierarchically;
- It does not use advices names to identify and clarify about the aspect itself, if it is an accounting, billing, transport, login, etc. aspect. But in the aspects encapsulation this can be done by grouping and organizing them, the notation does not prevent it.
- There is no dedicated modeling tool to model AO-BPM 2.0, or functionality in other modeling applications.

Figure 14 depicts an example of the process model from Figure 1 using the AO-BPM 2.0 Notation.

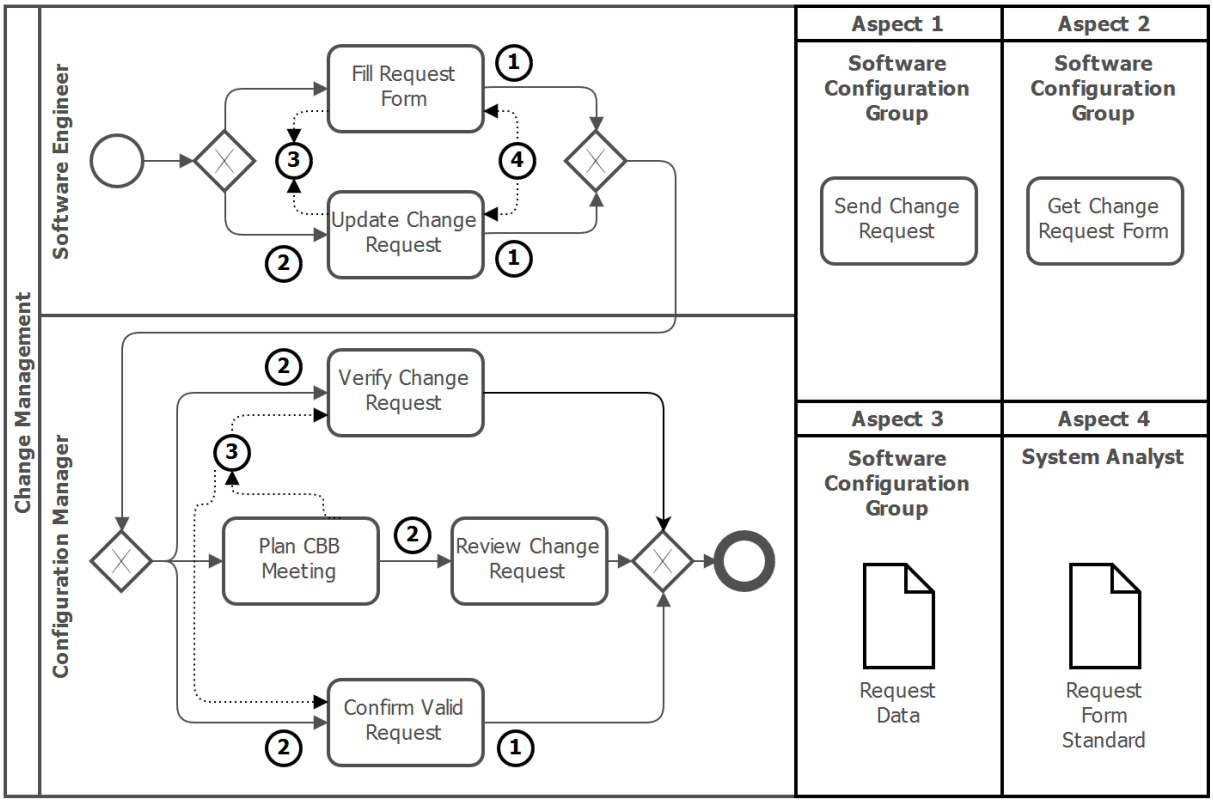

Figure 14: Change Management Process built using AO-BPM 2.0.

Detailed the four notations, will be presented three general topics relevant to the conceptualization, comprehension and detailing of aspect orientation: the conceptual difference between modularization mechanisms in BPMN and aspects; the problems with the traditional complexity evaluation in traditional modeling used in aspect oriented modeling notations; and when an aspect approach is recommended or not. Then the notations are compared.

3.5 Group, sub-process and aspect: are not they the same?

In BPMN: “A Group is a grouping of graphical elements that are within the same category. This type of grouping does not affect the sequence flows within the group. The category name appears on the diagram as the group label. Categories can be used for documentation or analysis purposes. Groups are one way in which categories of objects can be visually displayed on the diagram” (OMG, 2011). The group artifact provides a visual mechanism to group elements of a diagram informally and is used to highlight certain sections of a Diagram without adding additional constraints for performance as a sub-process would. It is used mostly for documentation, report and analysis and not, de facto, for modularization.

Group provides a semantical separation and categorization of concerns, and does not interfere with the objects or the flow. It is not an operational option for abstract syntax simplification (LA ROSA *et al.*, 2011), as a modularization, having only a concrete syntax purpose (LA ROSA *et al.*, 2011).

In BPMN: “A Sub-Process is a compound activity that is included within a process or Choreography. It is compound in that it can be broken down into a finer level of detail (a process or choreography) through a set of sub-Activities” (OMG, 2011). Sub-processes are addressed to granulate levels of information, condensing it in smaller objects with them inside. They are not primarily created to encapsulate concerns or represent crosscutting elements in a process model, even because modularization and separation of concerns are not the main objective of the BPMN modeling capacity. Modularization using sub-processes can provide benefits, in understandability, reuse, maintenance, error prevention, consistency and flexibility (LA ROSA *et al.*, 2011), therefore these benefits can only be applied at monolithic and function focused scenarios using pure BPMN, not gathering the scattered elements or transversal concerns (not only to a process, but a portfolio).

“Aspects are the elements designed to encapsulate crosscutting concerns and take them out of the core elements in a given specification or implementation” (SANTOS *et al.*, 2011). Aspect-orientation, in a way or another, involve the concept of modularization, and when the topic reached the conceptual level of BPM, especially in BPMN, the first association was sub-processes, as a mean to separate the aspects from the main process. The use of an aspect-oriented solution is not limited to the use of sub-processes elements and concepts, but the BPMN sub-processes elements and concepts cannot properly encapsulate and operationalize the aspect paradigm in a fully effective way, separating with the needed relevance the base process and aspectual process (SANTOS *et al.*, 2011).

3.6 Simplicity and Easiness of comprehension

Two terms in this undergraduate thesis are conceived to provide a better uptake about aspect-oriented business process modeling comprehensibility: simplicity and easiness.

Interpretation and absorption of information in a business process model are cognitive tasks. Two analogies in other scopes are provided to exemplify the concept of simplicity and easiness:

- Riding a bike is a simple task, but very complex at first attempt, after some experience and training someone can perfectly ride a bike and go from a place to

another. But, maneuver a bike is not so easy as learning how to ride it. To start riding a bike, just to simply use it as a transport, it is a lot easier them make maneuvers with one.

- Playing piano is a complex task, but very simple. Anyone just pressing the keyboard is “playing piano”, even if you see a video and repeat it in the piano it is “playing piano”. Play piano with quality is not an easy task, even for a beginner, but just press the keys and repeat mechanically a “Jingle Bell” is a simple activity. To a beginner piano player even the simpler scores are complex, and when the experience and training grow better and better is remarkable that a very extensive piano sheet, requiring less technique, is very easy, but not simpler because of its size; and a very short piano sheet, requiring too much technique, is very complex, despite the simplicity represented by its size.

Business process models can be measured by its simplicity and easiness in the matter of comprehension, often generalized by understandability research, addressing the complexity, simplicity, easy and compositeness as the same (SIAU e TIAN, 2008) (CARDOSO *et al.*, 2006) (TURETKEN *et al.*, 2016) (WAHL and SINDREAN, 2006) (VANDERFEESTEN *et al.*, 2007).

The easiness of a process model is based on the number of constructs related with the common appearance of them in usual models (kernel elements, the most used and easily available in most of the business process models) and the quantity of data a single element represents (condensing expressive capacity) (SIAU e TIAN, 2008) (RECKER, 2010).

> In most cases, the reason a modeling method becomes very complicated is it tries to incorporate as many constructs as possible so that it can model the domain complexity well. In doing so, the complexity of the real world will make a modeling method complex. Modeling method developers face a dilemma between expressive power and simplicity because simplified modeling methods are at risk of failing to model the complex real world due to insufficient constructs. To compensate for that, simplified modeling method tends to assign many denotations to one single construct (construct overloading). This creates problems for users to accurately interpret the modeling construct as the construct can mean a number of things in the real world. On the other hand, a complex modeling method with corresponding number of constructs to the real world, as recommended by the BWW ontology, is able to model the real world accurately. However, it will inevitably need to sacrifice simplicity to include many modeling elements and notational constructs in the modeling method. The resulting modeling method can become extremely complex and unwieldy. (SIAU e TIAN, 2008)

The simplicity of a process model is based in its size, abstract and concrete syntax (LA ROSA *et al.*, 2011) (LA ROSA *et al.*, 2011). In this aspect, the cognitive load and modularization is observed, after a number of elements in a process model the

comprehensibility tend to diminish, because the cognitive load is overstressed (OTTENSOOSER *et al.*, 2012).

Complexity, in a broad way, is addressed as a quantitative problem (CARDOSO *et al.*, 2006) and not properly qualitative. The use of an almost never used element, e.g. one of the many expressive and rarely used BPMN events, can harm the comprehension of a model (RECKER, 2010) in a qualitative form, not quantitative at all. Even if the overload of elements results in cognitive load, the use of non-kernel and mostly unknown constructs can lead to the same interpretation problem; in the first the reader need to reread or separate information, in the second he needs to stop reading and look for more knowledge to correctly absorb the information represented there. So, we cannot just infer that a model with ten elements, syntactically (abstractly and concretely) and semantically perfect, is comprehensible and a process model with more than thirty elements is incomprehensible.

The opposite of simplicity is complexity and the opposite of easiness is compositeness.

Combinations of simplicity and easiness, as conceived in this work, are exemplified:

- A simple and easy model represents basic elements with complete comprehensibility to the target audience in a size that does not hurt the absorption of information and with concerns quickly trackable and grouped.
- A simple and complex model represents few very well structured elements, but not-kernel and very rare in usual business process modeling, loading many semantical data in each element.
- A composite and easy model represents many not necessarily structured elements, but kernel, basic and iconic (SIAU e TIAN, 2008) ones, with fifty or more elements, organized in a linear flow and one directed flow or not.
- A composite and complex model represents many elements, hardly structured because of the number of them; not-kernel and not-basic constructs holding much concentrated information and showing a big number of elements.

In a scientific communication bias (FISKE, 2010) the redundancy of available and empiric business process models and their elements dictates the passive complexity, the reader is so exposed to these elements that it is already fixed in his memory. It is natural that common core BPMN elements (RECKER, 2010) (simple tasks, sequence flows, pools) are of easy comprehension, the readers see them in the business process modeling in a regular basis, but when an entropic element appears, like a BPMN overhead element (RECKER, 2010) (end

compensation, intermediate exception, conditional flow), the reader need to stop the cognitive process of reading the model and dedicates attention to study about these new entropic, not expected and new, elements. The latter case is an active complexity case.

To clarify the concept, we consider BPMN 1.2, released by OMG in January 2009, counting fifty-five objects. With the expansion of BPMN use (LIN e YANG, 2002) (ROSING, WHITE, *et al.*, 2015) and need for more expressive and significant elements (as marked business rules tasks) (RECKER, 2010) BPMN was updated to version 2.0, released by OMG in January 2011, summing sixty-one elements to the fifty-five, counting now a hundred and sixteen elements (WIKIPEDIA, 2016). We see many autodidacts and the small share of adequately trained BPMN modelers (RECKER, 2010), and it is hard to believe that most of the modelers community using BPMN know the hundred and sixteen elements shape and significance and more, all the syntax and semantic rules between them. When the modeler chooses only the most basic and core elements, he fails to take advantage of the expressive contribution of the notation (SIAU e TIAN, 2008). And even worse, when it hurts or change the notation formalism that affects its basic elements, syntax or semantics it is not considered anymore the core BPMN (OMG, 2011). It will not be recognized in respect of conformity (OMG, 2011), certification or certain modeling tools (LA ROSA *et al.*, 2011) (LA ROSA *et al.*, 2011).

These two categories, simplicity and easiness, are elicited here to clarify a statement: AO-BPM, AOBPMN and AO-BPM 2.0 help simplicity and hurts easiness. The good use of them makes the models simpler, but increases the complexity.

The benefits of helping simplicity is well known, operationalizing aspect-oriented modeling approaches (TAVARES e MARINHO, 2014) (SANTOS *et al.*, 2012) (VAN DE BERG, CONEJERO e CHITCHYAN, 2005) (CAPPELLI *et al.*, 2010) (CHARFI, HEIKO e MEZINI, 2010) (JALALI, WOHED e OUYANG, 2012) (WITTERBORG *et al.*, 2014) (SANTOS *et al.*, 2011) and corroborating with the modularization qualities to simplicity in business process modeling (MELCHER, 2012) (WEBER e REICHERT, 2008) (BALDAM, VALLE e ROZENFELD, 2014) (DUMAS *et al.*, 2013) (MENDLING e REIJERS, 2008) (DEVILLERS, 2011) (LA ROSA *et al.*, 2011). The size of the models gets smaller, the flow readability is improved and the redundant elements are encapsulated in advices, enhancing the maintainability, flexibility, agility and reuse.

The observance of hurting easiness is very obfuscated, because aspect-driven operations work with modularization or specific methodologies of it. Turetken et al. (2016) calls attention to abstraction and split-attention effect problems in modularity, sub-processes

might gain by abstracting less relevant information but losses in the cognitive load, incurred in browsing through and integrating fragmented pieces of models, counter-balancing the gain.

The analogy with reading a book is simple and illustrative, the continuous read of a book maintain the concentration and attention focused on the flow of content, e.g. a narrative. When the text flow presents a lot of cuts and references to other book elements (as Chapters, figures, tables, etc.) the concentration in the continuous task is broken, and the reader is urged to stop the cognitive process and search for the new element, proposed by the writer.

Turetken et. al. (2016) applied a questionnaire to business process analysts and modelers, and measured the understandability task effectiveness, understandability task efficiency, perceived usefulness for understandability and perceived ease of understanding. The analysis of the results is presented here:

a) The understandability task effectiveness factor revealed that flattened models lead to a higher effectiveness than the models where sub-processes are used. For local questions, modularization degrades effectiveness when overview + detail is used, i.e. where sub-process is shown separately, detached from its context. Likely explained by the split-attention effect, a browsing cost.
b) The understandability task efficiency revealed an interesting result. The number of correctly answered questions divided by the time spent for answering them is higher in the sub-processes.
c) The perceived usefulness for understandability revealed that fully flattened models are considered more useful in providing gains to user in terms of understandability in comparison with the models with sub-processes.
d) The perceived ease of understanding revealed that fully flattened models are easier to understand than models using modularization through sub-process. And any additional information on the process model can be perceived to increase the difficulty of understanding, this was mainly observed in the comparison between fully-flattened process models and the same models with grouping.

Turetken *et al.* (2016) worked with sixty practitioners, the majority had some degree of BPM knowledge but relatively limited familiarity with the BPMN.

The critics to Turetken *et al.* (2016) work are: Sub-processes in BPMN was not primarily created to improve understandability or comprehensibility, but to encapsulate and simplify blocks of elements abstracting them in a hidden (collapsed) or in an explicit (expanded) condensed representation; The group artifact is not created to modularize elements in business process models, only to semantically magnify information; it is almost obvious

that sub-process and its concepts are not fully understudied by a group of sixty practitioners without verified BPM or BPMN (focus in BPMN) knowledge, so they can freely respond a false level of BPM or BPMN notion; if the understandability (item d) of the practitioners are affected by an artifact that does not interfere directly in the semantic or syntax of the model so they do not know clearly the purpose of the group artifact or that the grouping was erroneously operationalized; item puts a stake in the rest of the work, why the understandability is so negatively affected using sub-process if they answered the average understandability task efficiency questions correctly about the process models using sub-processes?; BPMN has more than one sub-process syntax (Chapter 2.4.1), not addressed in this work; and the models used in the questionnaire are not available so we cannot know if the understandability problem was polluted by complex non-kernel elements mixed with the sub-process representation problem.

The separation of simplicity and easiness is specially made here to ground: the aspect-oriented notations presented in this work are not infallible or without problems, even if they bring simplicity, they hurt complexity. Still some more than others.

AO-BPM double hurt the easiness combining the main flow, crosscutting concerns pool and textual description. The reading starts in the main flow, if the main flow has pointcut so the crosscutting relation conducts to the advice (and the aspect), and if there are more than one crosscutting flow, it is necessary another information, the join point's detailed in the pointcut language. Only after the pointcut language and crosscutting relationship flow understood, the reader can return to the main flow and continue reading the process. It is a two-step split-attention distraction, and if the elements are complex or mixed in a hardened syntax context, the comprehensibility will be even worse. In the simplicity spectrum, the AO-BPM does not guarantee the diminishment in the number of elements in a process model, in this case does not used any benefit from the modularization, only the segregation of the process aspects (that could be achieved using the group artifact).

AOBPMN solved the double hurt problem in attention, but introduced a new problem, the crescent number of elements. The advices in AOBPMN are not composed by a single element (e.g. just an activity), like AO-BPM, following the classic expanded sub-process structure. Taking an isolated activity of the main flow and make it an aspect, will include a total of three elements (not just changing the element semantic and position): two conditional events (one to illustrate the pointcut in the main flow and one in the aspect itself, starting it) and a final event, not to mention the very activity that now goes to the encapsulated advice. Looking at Figure 1, using BPMN, and Figure 13, using AOBPMN,

even using the modularization concepts AOBPMN presented more elements than BPMN (counting the pointcut, initial and end events in the aspects. Twenty-two in BPMN against thirty-one in AOBPMN), improving the compositeness of the process model, without necessarily improving the easiness. A proposal to decrease the number of elements and comply with BPMN is just made by eliminating the initial and final events inside the advices, to preserve the most of the simplicity, because the notation permits a sub-process without these objects, just with one activity.

From the three-analyzed aspect-oriented notations AO-BPM 2.0 presented the best solutions for simplicity and easiness. The circled elements as pointcut facilitate its placement on the model and its organization. They eliminate the need to join points expressed in other objects, as textual annotation in AOBPMN and ground elements combined with crosscutting relationship and pointcut language in AO-BPM. It is intuitive for the reader to look e.g. at the circled number “3” and associate with the pool (advice) containing the aspect 3. AO-BPM 2.0 also registered an increase in the elements number, comparing Figure 1 and 14, lucidly caused by the pointcut number. The circled numbers help simplicity in the relation between advice and main flow, and the complexity increased just for the one step of attention, caused by the conduction of the flow from the pointcut to the advice and vice-versa.

Although the simplicity and easiness analysis, the main goal of the aspect-oriented modeling notations are not primarily the comprehension and facility to read, however an indiscernible model is useless, using the best modularization or not.

3.7 When should an aspect oriented modeling approach be used?

The large number of modeling languages available make it difficult to select the most suitable to use or to improve them. The language need to be appropriate related with the needs, and many of them define overlapping concepts and the usage areas are narrowly specific or broad (WAHL and SINDREAN, 2006). BPMN has attained a great level of consensus and standardization, supported by dozens of commercial tools and used by thousands of practitioners across a wide range of industry verticals, are perceived by the adopters with a suitable tradeoff between instrumentality (usefulness and performance of BPMN for process modeling) and ease-of-use (complexity of creating BPMN models) (DUMAS and PFAHL, 2016). On the other hand, BPMN have some problems handling routine work, knowledge work, data modeling, supporting business rules or decision making (in progress with DMN) and organizational modeling (WIKIPEDIA, 2016).

Aspect-oriented Modeling Notations (AOMN) presented in this work is extensions or symmetric modifications/additions to BPMN, then they share the same limitations and benefits of it. However, the AOMN have a narrow scenario usage than traditional BPMN, because the central focus of them is deal with an aspect-driven scenario and not general business process cases.

The operationalization of an AOMN in a non-favorable scenario can lead to a not suitable solution for this specific necessity of modeling and representation of information. To correctly use this solution an expertise and understandability of BPMN are needed, as the first complement the second with specific syntax and semantic to best represent aspects in a process model. First, it comes the know-how in BPMN, and then in AOMN.

Use of an AOMN solution in a business process portfolio without aspect, crosscutting or soft-goal appeal hamper the simplicity and complexity, as already seen in Chapter 3.6, and the use can increase the number of the elements in the model, because of the little scope and the fact that was just one business process. An environment without aspects, or few, is not an appropriate scenario for an aspect-driven solution.

A business process modeling using AOMN cannot be done without guidance of a business analyst or specialist, the aspects, crosscutting concerns and soft-goas are elicited by the organizational analysis, to guarantee the effectiveness and conformance of the artifact built. A business process analyst cannot define advices without knowing what is and what is not aspects, crosscutting concerns and soft goals from a business process description, that cannot be done just by “hinting” or “guessing”, e.g. segregating all the most repeated elements and calling them aspects.

An average cost benefit solution for an AOMN operationalization is seen in scenarios: with big amount of business process consistently modeled; with constantly repetition of concerns in business process models; environments where the aspectual and non-functional notions are clear and well defined; large repositories of business process models needing modularization and clarification of orthogonal interests; flexibility is required; adaptability is required; the depiction of interests is required, e.g. separation of functional and non-functional elements for business analysis; modularization of processes with enormous number of elements, e.g. hundred, when BPMN modularization solutions is not an option; and so on.

3.8 Aspect-oriented modeling notations comparison

Despite the differences between the notations to represent and give significance to the concept of aspects, they follow clear ideas of aspect oriented programming (AOP).

Table 1 aims to clarify and compare some differences between them.

Table 1: AO-BPM, AOBPMN & AO-BPM 2.0 comparison and differences

	AO-BPM	AOBPMN	AO-BPM 2.0
Pointcut	Ground element	Intermediate event object	Contoured element
Join Point	Crosscutting relationship	Text annotation artifact	Directly associated
Advice	In the aspect language	In the text annotation	Graphically explicit
Aspect naming	Not addressed	Directly in the aspect	Not addressed
Precedence	In the aspect language	In text annotation	Graphically defined in the elements order
Encapsulation of aspects (aspects in aspects)	No	No	Yes
Modeling tool to represent?	No	Yes, any BPMN modeling tool	No
Functionality in modeling tool(s) to represent?	Yes, Oryx editor	Yes, any BPMN modeling tool	No

Source: Elaborated by the author

After analyzing these three notations for aspects graphical representation in conceptual modeling of business processes, AO-BPM 2.0 was chosen to be more thorough and detailed in this undergraduate thesis, complementing the work of Tavares and Marinho (2014). AO-BPM 2.0 has advantages and disadvantages over the others, listed below:

- Advantages:
  - Presents simple and intuitive constructs dedicated to aspect representation (SIAU e TIAN, 2008);
  - Enables scalability;
  - Does not pollute the model in scenarios with many pointcuts, join points and advices;
  - Simplicity in syntax and semantics makes it easier to modifications and improvements;
  - Easier maintainability, just insert or delete contoured elements in the flow and crosscutting concern pools;
  - Encapsulated aspects are externally targetable;

- Allows the hierarchy of aspects and the use of them for other aspects;
- Works with BPMN modularization mechanisms, not limiting it;

- Disadvantages:
  - Needs a modeling tool, like any BPMN modeling tool for AOBPMN, or a built-in feature, like Oryx Editor for AO-BPM;
  - Does not has a static or dynamic weaving service;

After analysis of these items and observance of its advantages, principally in conceptual modeling, AO-BPM 2.0 was chosen to be better analyzed and then operationalized in the case study. This work does not try to solve all the drawbacks or disadvantages of AO-BPM 2.0 presented on it, the proposed improvements and analysis made are dedicated specifically for the case study, in view of conceptual modeling; although can be used in other scenarios.

4 AO-BPM 2.0 analysis

Noting the characteristics and qualities of AO-BPM 2.0 to represent aspects in conceptual models, dedicated to BPM initiative (RUSSEL e VAN DER AALST, 2016), we discuss in this Chapter a complementary and more elaborate analysis of it, continuing Tavares and Marinho (2014). The goal of this Chapter is:

- Clarify its symmetrical position in relation to BPMN, why it cannot be an extension of BPMN;
- Observe the restrictions and indicate good notation practices;
- Align the notation to proposed modeling formalism (RUSSEL and VAN DER AALST, 2016), demonstrating that its modeling capability is effective.

4.1 Symmetry with BPMN: why AO-BPM2.0 is not an extension to BPMN?

BPMN (OMG, 2011) allows the domain modeler to add non-standard elements or Artifacts to satisfy specific needs, and still valid in BPMN core. Extension attributes must not contradict the semantic of any BPMN element. If it is a mandatory extension, the compliant implementation must understand the extension, if it is an optional extension so the complaint implementation may ignore the extension. Each BPMN object has its own syntactical and semantical rules, and the alteration must not alter these aspects of the basic flow elements (events, activities and gateways).

The formal rules allow the creation of extended objects in BPMN as artifacts, maintaining its compliance with the core notation. The artifacts created this way need to comply with specific BPMN artifacts rules, and AO-BPM 2.0 (and AO-BPM) does not do it. Each one of the flow objects has a well-defined connection responsibility in BPMN (CAPPELLI *et al.*, 2010) and AO-BPM and AO-BPM 2.0 do not comply with this responsibility.

The pointcut and join point in AO-BPM and AO-BPM 2.0 are different from each other. AO-BPM (CAPPELLI *et al.*, 2010) pointcut is operationalized by a new connector object, the crosscutting relationship, connected in an element in the main flow, as they say: “Graphically, the join points can be represented by a ground element, which locates near it, allowing that the source of the crosscutting relationships be the crosscutting element, and the target be the ground element representing the join point”. The join point is described in the pointcut language, the complementary information about the aspects shown. In AO-BPM 2.0 (TAVARES and MARINHO, 2014) they describe the pointcut and join point as: “The aspect identification in the business process model is made using an identification number in a circle attached to the flow. The circle junction with the number will be the artifact that denotes a join point”. The AO-BPM 2.0 objects can represent, at same time, the pointcut and join points, ordering the pointcut in the flow (do not need the “before”, “after” or “during/around” description, cause the number position already abstract that representation) and associate the numbers with the aspects pool, identifying them.

The ground element implemented in AO-BPM is nothing more than a connection artifact, as Cappelli (2009) claims that they are using a symmetric approach to BPMN and the extensibility of objects creation are through artifacts. The circled number in AO-BPM 2.0, as the authors address them, is tracking artifacts, to identify the respective aspect. BPMN specification (OMG, 2011) strictly says that: an artifact must not be a target or a source of sequence flows; and an association is used to link information and artifacts with flow objects (events, activities and gateways). And these two statements remove the possibility of those two notations to be extensions of BPMN.

They are not extensions of BPMN, but they are symmetric to it, as the symmetric strategy does not define a specific abstraction to distinguish crosscutting concerns and basic elements. Both are represented using the same abstraction. The difference lies in the way that a crosscutting concern is comprised with the basic elements. Crosscutting elements are composed with basic elements through a specific composition mechanism (CAPPELLI *et al.*, 2010).

AO-BPM 2.0 has a strategically advantage over the other notations presented in this work, in addition to an expressive and precise way to combine pointcut and join points, it permits the aspects to be targeted by connection flows. This characteristic may seem irrelevant at first thought, but Figure 15, 16 and 17 display that problem.

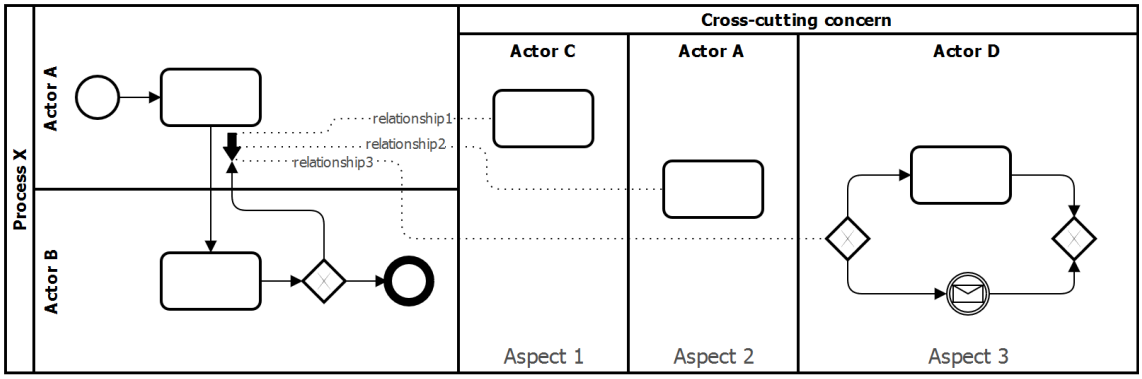

Figure 15: AO-BPM target problem example.

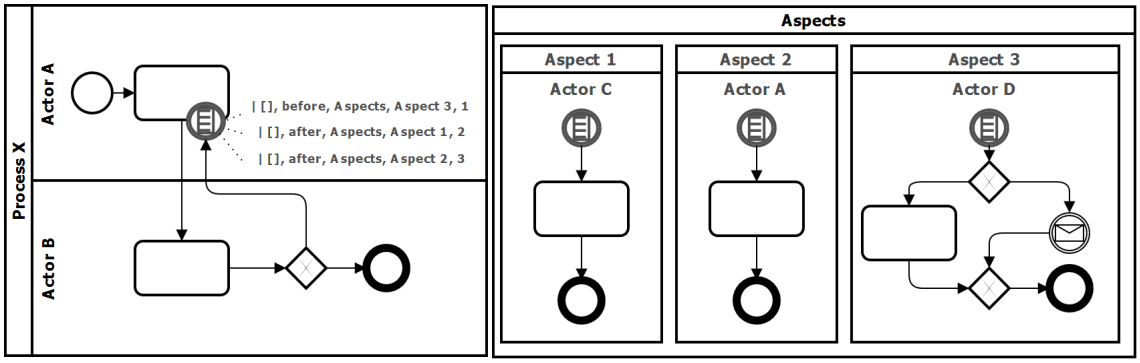

Figure 16: AOBPMN target problem example.

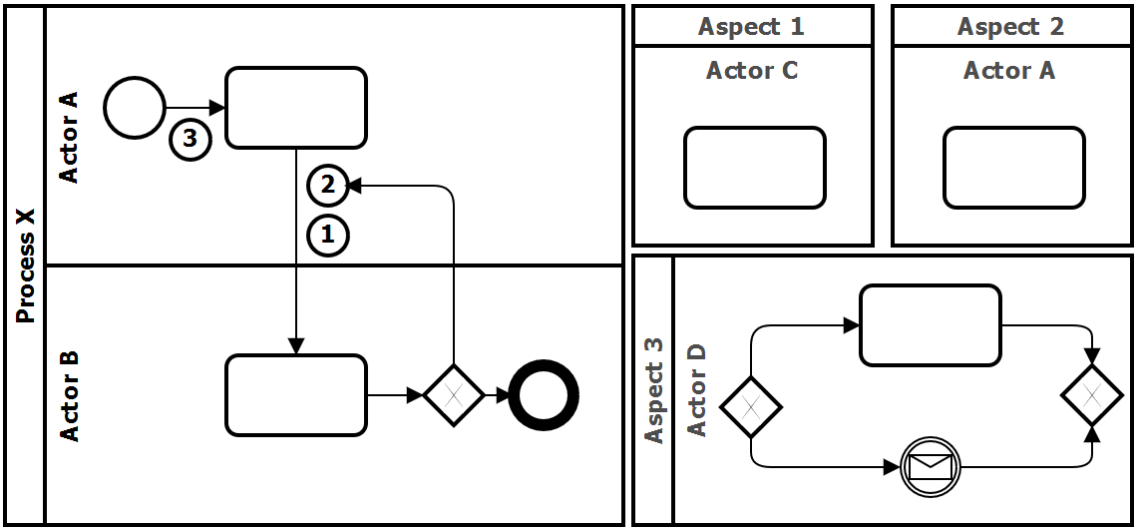

Figure 17: AO-BPM 2.0 target problem example.

The pointcut in Figure 15 and 16 can be targeted, but the aspects cannot. In Figure 15, 16 and 17 the flow present the probability of returning for a previous task, a loop, and the intended target is the aspect 2, in all the three cases. In AO-BPM, Figure 15, it is the worst scenario of the three, without the pointcut language detailing the join points we cannot even know if the sequence flow returns before the first task, Aspect 3, or after it, Aspect 1 or 2.

In AOBPMN, Figure 16, we can see the join points, but not the aspect addressed in the sequence flow, because it only connects with the pointcut. This operation in AOBPMN is

impossible, because it hurts the formalism of BPMN (attached event being targeted by incoming sequence flows) and is incompatible in any way.

In AO-BPM 2.0 we can address the aspect itself, because the join point and pointcut are very clear and atomic in that concept, e.g. in Figure 15, AOBPMN, the sequence flow targeting the intermediate conditional event (AOBPMN pointcut) cannot clearly say what is the next step of the process, aspect 1, 2 or 3; and in Figure 17, AO-BPM 2.0, the aspect is clearly targeted, the next step of the process is the aspect 2.

This target problem may invalidate or limit several realistic scenarios in which aspects simply become not targetable, in a gateway or message flow (in a scenario with more than one pool) a flow cannot return or reach an aspect, so we cannot implement aspects in scenarios with message flowing or with need for returning into them (e.g. Figures 15, 16 and 17) without this atomicity of data, do not provided in AO-BPM and AOBPMN. Future changes in notation; removing aspects from the encapsulation (on the other hand hurting the aspect approach); repeating part of the process; labeling the sequence flow (in the AOBPMN case); etc., are examples of remediation to solve these syntactical problems.

4.2 AO-BPM2.0 syntactical and semantical rules: AO-BPM2.0 limits

Considering the symmetry with BPMN, some details about the elements ask for clarification and detailing, intending to preserve and do not harm the process model analyzability, comprehensibility, consistency, flow and readability. The items are listed below:

a) Initial and final elements of the control-flow must not be aspects;
b) Aspects must not contain the final element of a control-flow;
c) Join points should not represent the beginning or end of the process, whether intended or not;
d) Gateways must be able to ensure the continuity of the logical sequence of activities defined by the process;
e) Gateways must be initiated and terminated inside the aspect that do not adversely affect the interpretation of the model, since a gateway can be started within the aspect and its junction can be finished out of it, leaving a gateway free from convergence element released by the model;
f) Link events must not connect encapsulated elements with main flow elements, and vice-versa;
g) Link events must not connect two different encapsulated aspects;

h) The control flow that take place inside the aspect must not give margin to external structuring elements inside it;
i) If the aspect encapsulates a control-flow, then the quantity of sequence flows before the encapsulation must be the same of sequence flows after the encapsulation;
j) Boundary events must not be aspects.

Business process models need modeling formalisms to capture and communicate business process, two of them are comprehensibility and analyzability (RUSSEL e VAN DER AALST, 2016). Comprehensibility handle the presentation of details captured in a form that is intuitive to users, retaining the resemblance to the existing operational process, the analyzability facilitates further analysis to stablish its static design-time correctness as well as allowing for monitoring of its operation at run-time.

Conceptual and analytical simple problems can be observed about the wrong operationalization of AO-BPM 2.0. For example, initial and final elements have semantical and analytical value, initial elements spawn the first token in the process model, representing the beginning of an instance, and final elements catch (one or many) the incoming token(s) to end the current process, representing the end of an instance. Encapsulating the initial and final elements of a process (or sub-process) conduct the readability and analyzability to errors.

To be susceptible to BPMN practices and methodologies, e.g. analysis, enactment, elicitation, etc., the resources must be in business process compliance with this notation, or be an extension of it. AO-BPM 2.0 is not an extension according to it formally, so, it prescribes their own limits aiming the alignment with all the other notation aspects, despite the syntactical and semantical modified concepts. These syntactic and semantic details seek to allow AO-BPM 2.0 use as much as possible of the practices, methods and guidelines of BPMN.

4.3 Quality evaluation and formalism of AO-BPM2.0

Russel et. al. (2016) determined eight qualities for model formalism, to effectively capture and communicate business processes, they are repeated here and each of them related to the AO-BPM 2.0:

- **Expressiveness:** It needs to be able to capture the complete range of concepts that occur in the domain of interest. AO-BPM 2.0 emphasizes the expression of aspects, the main concept of an AOMN is representing these elements, giving them primary expressive relevance.

- **Suitability:** The range of modeling constructs that are available should mirror the concepts and needs that arise in practice. AO-BPM 2.0, as already said in Chapter 3.7, is suitable for an aspect-favorable scenario. Representing business process without considerable aspect-needed solution can probably not be the right or best solution. An association is made with modularization in BPMN, using the concept of it in small business process models do not really bring any help, just easiness is hurt (TURETKEN *et al.*, 2016).
- **Sufficiency:** There should not be an excess of modeling constructs, such that the same underlying concept can be represented in many ways. In terms of aspect representation, it is like a global task or global process BPMN syntax and semantics (OMG, 2011), but instead of using thick edge use numbers with contour. The BPMN modularization concept, as discussed in Chapter 3, cannot express all the aspectual necessities of the chosen domain. Aspect-oriented dedicated modeling constructs can be used with modularization ones, their concept is the same, but denotative meaning not.
- **Precision:** It needs to be able to capture concepts occurring in the domain in a precise and unambiguous way. Some notations are deliberately vague or have semantics that are intuitively clear in most cases, but not if they are used in an atypical manner. AO-BPM 2.0 does not have a weaving service conducting the analysis of implementation of the core process and crosscutting concern process. The semantic analysis presented in this work deals only with the construction and design of conceptual models. The way pointcut, join points and advices are operationalized in AO-BPM 2.0 enables analysis by token pass (OMG, 2011), as proposed by the specification form of BPMN (OMG, 2011) as the token traverses the sequence flow and instead of going straight to the next activity it accesses the aspect by pointcut in the sequence order, performs the crosscutting concern activity or process and then returns to the flow, to the aspect or activity.
- **Enactability:** There should be sufficient detail to allow the business process to be directly enacted without requiring the elicitation of any further information from users. AO-BPM 2.0 can use global task and global process idea to permit enactment of its models, avail the benefits inherent for flexibility, reuse and crosscutting concern modularization.

- **Comprehensibility:** It needs to present the details of the business process being captured in a form that is intuitive to users. The captured process model should retain a resemblance to the operational process as it exists in practice. It is a bit subjective, depending on training, experience and taste. The comprehensibility, compared with traditional BPMN, is hurt because of the need for learning more syntax and semantics just for the aspect-orientation. But helps the reader to separate the concepts, soft-goals and non-functional elements depicting more clearly the value, functionality and goals of the modeled business process.
- **Analyzability:** It should facilitate further analysis of the business process both to establish its static design-time correctness as well as allowing for monitoring of its operation at run-time. Some notations are easier to analyze than others, in BPMN models often are needed a conversion to Petri Nets or some other formalism providing analysis techniques. AO-BPM 2.0 presents two problems concerning this item: i) AO-BPM 2.0 with its symmetrical relation with BPMN inherits its semantically dubious problems (VAN DER AALST, 2013), making it (as BPMN) not suitable for formal analysis; ii) To formally verify BPMN models, the use the Model Driven Engineering (MDE) approach to transform a BPMN model into a Petri net model are observed. The MDE approach provides the tools, concepts and languages to create and transform models based on their meta-models and transformation rules (MOULINE e LYAZIDI, 2013), BPMN has automatic mechanisms and tools for transforming automatically into Petri Net, then its analysis can be performed automatically, starting from a model using it. AO-BPM 2.0 in addition to not have a dedicated modeling tool also does not have this transforming artifice, damaging its potential to analyzability. This does not preclude the analyzability of AO-BPM 2.0.
- **Independence:** The formalism should not rely on any specific technological foundation. AO-BPM 2.0 fully attends this item.

Generally, AO-BPM 2.0 can respect all these modeling formalisms, presenting most difficult in analyzability, due to semantical problems of conceptual models (VAN DER AALST, 2013) inherited from BPMN, and comprehensibility, caused by the necessity of learning a new syntax and semantic to understand it. Expressiveness, suitability and sufficiency depend on the use of the notations and the modeled domain. Enactability needs more empirical research. Precision and independence are correctly addressed.

4.4 AO-BPM 2.0 proposals for improvement

This chapter addresses mechanisms for improvement of AO-BPM 2.0 operationalized in the case study, in Chapter 5. Aiming to maximize the representation of aspects and cover some topics and disadvantages of this notation these improvements were developed.

This work, where the proposed improvements are put into practice, shows the hierarchical decomposition of seven business process models, representing the claimant's care through the Secretariat Administrative Request (SAR) at the highest level (Figure 28) the macro process, the items contained in the SAR explaining the processes at the intermediate level (Figure 30) and five of the latter on the lower level (Figures 32, 34, 36, 38, 40), atomic processes.

4.4.1 Crosscutting concerns and general aspects operationalization

Framing primary objective, here, the concepts and situations from which potential issues could arise were widely analyzed. An extension of AO-BPM 2.0 introducing a new element to the notation is suggested: a crosscutting process pool of scattered and general aspects. This improvement concerns to the symmetry with BPMN, without creating new syntax or modifying the current. In the low-level processes, significant amount of aspects is found, but also a redundancy among similar aspect models, not taking full advantage of the modularization concept with too much elements repeated and scattered.

There were not only crosscutting concerns in one model as well as crosscutting aspects permeating many models. When abstracting the crosscutting concerns like aspects in a detached lane of the model, and when, in a process portfolio, various aspects abstracted in lanes repeat among brother processes? The concept of modularity is despised if, beyond modeling a specific model, cannot modularize a group of them, presented in a portfolio or modularized inside a macro process. For that purpose, the general aspects pool is added up, Figure 18 exemplify one.

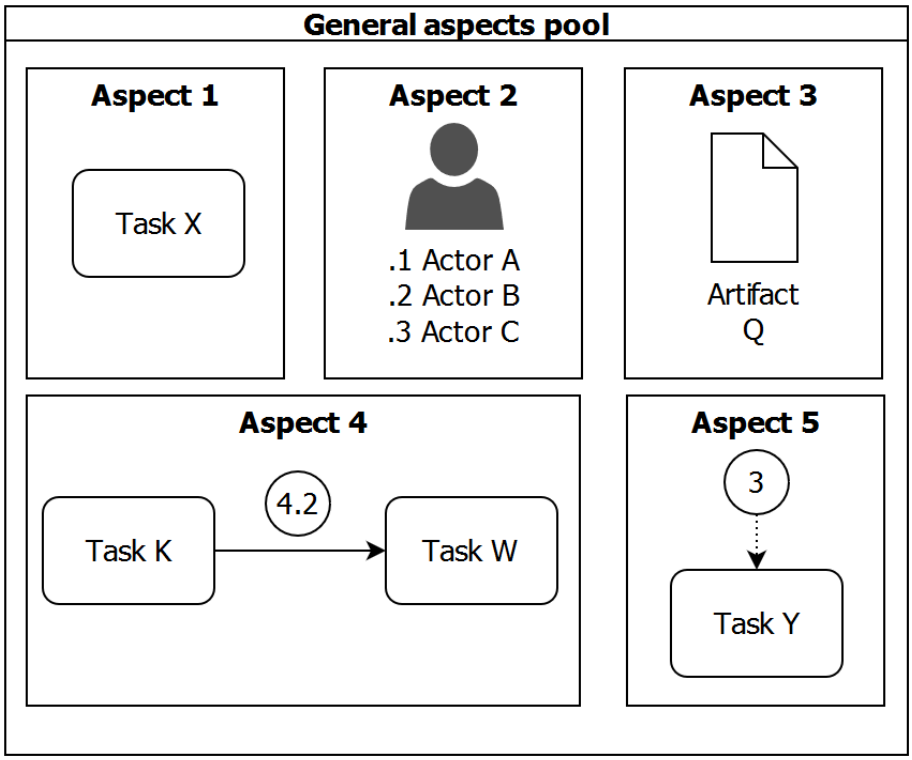

Figure 18: General aspects pool example.

In Dumas *et al.* (2013) the concepts of embedded and global tasks and processes are presented. Embedded can only be invoked locally by its parents' processes, from which they derive, and global can be invoked by any process that requests it. Using that same concept in this work as global and embedded aspects, determined aspect that permeates many processes, with the same concept, will be a general or overall aspect, while specific aspects of those processes will be embedded or global. A process, then, may: have no embedded aspect if their aspects are common to all the others included in the framed macro process, only general aspects (Figure 33); have only embedded aspects in case all of its aspects do not repeat or its concept is the only one in that context, only particular aspects (Figure 31, exempting the artifact Aspect 1).

The more aspects eligible as global arise, the better modularization and gain of the general aspects pool. In the reverse situation, the use of a new pool only for aspects culminates in the need for a new model as a repository only for the same, thus in scenarios where the general aspects appear in only a minimal number of processes, in the macro process, the pool creation is not recommended. For example, in a scenario with a macro process that contains one hundred sub processes and generic aspects permeate only ten and not all at once (some subsets), it is not recommended to overload the interpretation with one more model, which would be the new pool, in this instance it is more efficient to embed all of them.

General aspects are identified by ascending order numbers, positive integers, while the local aspects to the processes are identified by rising Latin Alphabet letters, starting in *a*. That operation is made rather by the greater number of global aspects than the embedded aspects and the method of order in each scenario should be analyzed properly.

The general aspects pool serves as an index of aspects to the macro process, it is recommended its presence every time one analyses and evaluates the models that use this tool. It does not directly affect easiness since the reading of the global aspects accompanies the reading of the embedded aspects. The reader does not need two steps or two different information, just the global aspect pool together.

4.4.2 Redundant resources operationalization as aspects

Observing the example in Figure 33, three resources as aspects are noticed: the secretariat, the direction and the faculty. They perform similar tasks and activities and operationalize similar artifacts, becoming aspects regarding this information. There is no specific syntax exposed in AO-BPM for aspects including resources, even if they are redundant in the concepts involved.

Analyzing the relationship between a resource and an aspect numerically, if an A resource is involved in N aspects, A x N aspects are seen. Inductively, case X amount of resources involved in a Y number of aspects, X x Y aspects are seen in order to address all aspects relevant to their resources. Illustrating based on Figure 4, if the secretariat, the direction and the faculty (who perform in various process aspects 3, 4, 5, 6 and 7) were not operationalized as aspects then there would be redundancy of aspects based on resources performing them. There would be fifteen aspects needed, five for the office, five for direction and five for the faculty in the process of representing the necessary resources to perform their proper aspects. In this proposal, only six aspects are needed, one aspect to sub-categorize the resources and one aspect to each sub-categorized resource. Fifteen aspects reduced to six, in a small setting like the one shown. In a similar scenario where, e.g., ten resources and thirty aspects take three hundred to represent all, if it is used the approach proposed in this thesis, it can reduce that number to thirty-one, the amount of N + 1 sub categorizer resource aspect.

The resources on Aspect 2, Figure 27, are an addition to syntax, not a modification of it. It represents the sub categorized aspect with a suffix identifying it, rather than a single numeral, e.g., representing the direction resource in Aspect 5 with ".2" suffix, their respective aspects end with a ".2" representing a direction aspectual activity.

4.4.3 Change in syntax to enable easiness of comprehension

Tavares and Marinho (2014) proposed a simple syntax to graphically represent pointcut and join points, the circled numbers. The circled numbers represent all the advices, independently if it is an embedded advice or the global advice, proposed here.

The following modifications aim to reduce the burden on the interpretation of the syntax and facilitate the recognition of objects. The pointcut representing the global aspects are illustrated by numbers contoured by squares, while embedded local aspects are illustrated by the traditional circled number. Figure 19 exemplify the proposal.

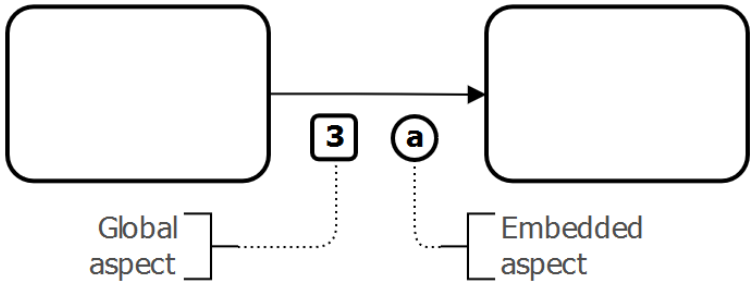

Figure 19: Proposal of change in the AO-BPM 2.0 syntax.

This mechanism to enhance the syntactical expressivity slightly increases the amount of abstract information, hurting the simplicity of the circled number, but helping the aspect identification and contextualization. With this graphical distinction, the modeler can use numbers in the global aspects pool and numbers in the embedded aspects pool. Figure 27 uses letters and numbers because that is the actual operationalization in this work.

That is a symmetric proposal to AO-BPM 2.0, do not altering its use or abstraction.

4.4.4 Aspect determination by value adding analysis

In the method proposed by Tavares and Marinho to discover aspects they recommend the depiction of between five or seven activities to classify them as “important”. The third step of the aspect elicitation (establish the scope of the processes) is a more specific one and is directed to analyze the process in a more restricted context, identifying the potential aspects of the process. They emphasize the importance of contextualizing the analyzed business process and its importance in the organization before move on the next step of the method, but do not provide means to clearly measure this “importance”. Value analysis described by Dumas et al. (2013) is proposed to enlist this “between five and seven important activities”.

To classify and quantify the value of a step, activity or task there is a technique whereby an analyst threshes a process model, extracts every process step and analyzes it based on its value (DUMAS, LA ROSA, *et al.*, 2013). It can be done, for example, in

accordance to the organization value chain, used to see how business works to create value, and where some of the cost drivers and challenges are (BYTHEWAY, 2014).

The steps, activities or tasks can be separated as value-adding, business value-adding and non-value adding. Value-adding is a step that produces value or satisfaction vis-à-vis of the costumer. Business value-adding is a step necessary or useful for the business to run smoothly, or it is required due to regulatory environment of the business. Non-value adding is a step that does not fall into any of the other two categories. Classifying steps into some of these three is to some extent subjective and depends on the context (DUMAS, LA ROSA, *et al.*, 2013).

In Dumas *et al.* (2013), a repairing washing machine process example is provided. The steps when the technician diagnoses the problem and repair the machine are value-adding, the steps about recording the defect are not directly related to the core washing machine fix, but recording defects and their resolution helps the company to build a knowledge base of typical defects and their resolutions, this is business value-adding.

Using this technique, more than aspects can be identified, also disposable activities, steps or tasks. Non-value adding, business-value adding and value-adding are, progressively and respectively in this order, candidates to be aspects or soft-goals.

Next Chapter presents a case study that illustrates the application of the extended AO-BPM 2.0 in comparison with AO-BPM and BPMN.

5 Case study: aspect orientation in a real setting

The scenario chosen to evaluate the application of BPMN, AO-BPM, AO-BPM 2.0 proposal was the processes from the Secretariat of the Information Systems School of a public university in Brazil, the Federal University of the State of Rio de Janeiro. Those processes represent the administrative services provided to the students (CARVALHO, SANTORO and CAPPELLI, 2015). All of them start with a student filling a standard form specifying his requirement. These are not functionally involved in the mission and vision of the organization showing great probability of having non-functional crosscutting concerns, i.e. aspects.

The goal of this case study was to compare the two versions, BPMN and AO-BPM 2.0, in terms of understandability (easiness to be learned by users) and representativeness (possibility to represent aspects in a broader sense). The processes were already modeled using the BPMN and AO-BPM in its first version, and then the same processes were modeled using AO-BPM 2.0. In order to measure these dimensions, seven processes were modeled with both notations and applied the following metric is used: LOC Metric (CARDOSO *et al.*, 2006) to analyze complexity and compositeness in models with AOMN. The results analysis is presented in Chapter 6, for discussions.

In the Information Systems School at UNIRIO, interactions between a student and the Secretariat Office are based on a form called Secretariat Administrative Requirement (SAR). Through it, a student can require a service or product, in that case a document. The student selects one or more desired item(s) and the process starts. It is important to notice that since those processes are very correlated to each other, since they deal with similar objects and goals, they constituted a good set of models to apply the aspect oriented approach.

These next Chapters present the details of process models and the aspects identified within them, which were modeled in both notations. Chapter 5.2 presents the process models in AO-BPM. Chapter 5.3 presents the process models in AO-BPM 2.0.

5.1 LOC Metric and elements counting

The LOC Metric is a broadly measure form used to analyze complexity (CARDOSO *et al.*, 2006). The complexity in this way is not the complexity addressed here, but the simplicity, defined in Chapter 3.6. The LOC Metric simply counts objects in the model, inferring the compositeness from that data, the number of flow elements in a model hurts its comprehensibility (MENDLING, REIJERS e VAN DER AALST, 2010), greater number of elements greater the probability of errors in the model (LA ROSA *et al.*, 2011). As the AOMN proposes the simplification of the business process model as one of the benefits, the LOC Metric is suitable for that kind of analysis.

Divided in three categories: “If we view a process activity as a statement of a software program, we can derive a very simple metric (metric M1) that merely counts the number of activities (NOA) in a business process” (CARDOSO *et al.*, 2006). So, NOA counts the number of activities in the process model. NOAC (Number of activities and control flow) count the number of activities and control-flows, providing a structuring qualification, e.g. the split objects requiring correspondent join objects. NOAJS (Number of activities, joins and splits) count the number of activities, joins, and splits in a process for languages or notations not well-structured.

AO-BPM, AOBPMN and AO-BPM 2.0 can be evaluated by the simplicity using NOA, because they are symmetric or an extension of BPMN and do not interfere in the syntactic structure. An adapted NOAJS is used here for counting criteria, so activities, data objects, splits and joins are then counted.

The act of introducing aspects in each of the notations has its own mechanics, including the number of elements in the model. As in BPMN, the number of elements is proportional to the probability of errors, in AO-BPM and AOBPMN the amount of aspects requires more space and more work to fit the constructs of the operationalized aspects in models such as the crosscutting relationships in AO-BPM, needing just to add a contoured element in AO-BPM 2.0.

Every time an aspect is identified and encapsulated in each notation the following situations are met:

- In AO-BPM a new crosscutting concern pool is created (only for the first aspect) and the aspect is framed in a lane, if the aspect is a data object connecting to other aspects so the simple association flow is provided between them, if not the aspect

need to relate to a ground element (pointcut) in the core process model, using a crosscutting relationship, and this crosscutting relationship need a name in the model. The join point and advice is provided by the pointcut language, a meta-model explaining the complementary information about the aspect, not explicitly provided in the crosscutting concern pool. So, a generic crosscutting concern pool (and lanes), a ground element, a crosscutting relationship and a new separated pointcut language pool is required, in the case of just one aspect connected with just one pointcut.

- In AOBPMN a new crosscutting concern pool is created for each aspect category (e.g. a pool for security aspects, a pool for login aspects, a pool for sustainability aspects, etc.) and the aspect is framed in a lane, receiving an identification name (to enable the precedence), in the core process model an intermediate conditional event (pointcut) is attached to the activity before, after or around the aspect, for each aspect invoked before, after or around this attached element an text annotation is created, attached to the intermediate conditional event, explaining the join point and advice. So, a specific and identified crosscutting concern pool (and lanes with identification names for the join point and advice), intermediate conditional event, textual annotation artifact and dedicated aspect lane name is required, in the case of just one aspect connected with just one pointcut.
- In AO-BPM 2.0 a new crosscutting concern pool is created for each aspect (with the aspect framed in it), this pool receives an identification element (e.g. number, letter, etc.), a contoured element (pointcut) is created in the core process model, placed in the exact position close to the sequence flow that the aspect will be invoked. So, a crosscutting concern pool (with the identification element) and a contoured element are required, in the case of just one aspect connected with just one pointcut.

The LOC Metric is used in this work to measure the simplicity impact of the aspect oriented operationalization, with AO-BPM 2.0 and BPMN, assessing the results on each of the models with an adapted NOAJS.

5.2 AO-BPM operationalization

The whole set of processes modeled is composed by: Interaction with the secretariat, selected items index, requirements breakdown, course completion statement, temporary course enrollment drop, test review and exemption or utilization of discipline.

Tavares and Martins (2014) identified the aspects in those processes. Figures 20 to 26 show their representation in AO-BPM.

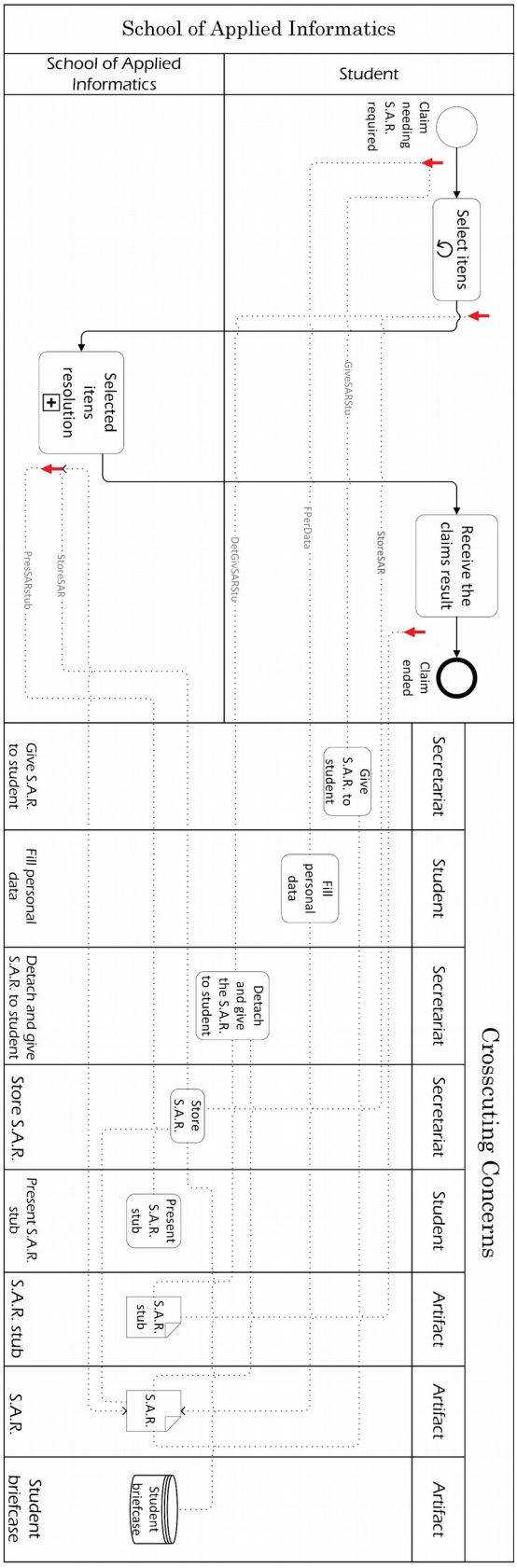

Figure 20: Interaction with the Secretariat Process built using AO-BPM..

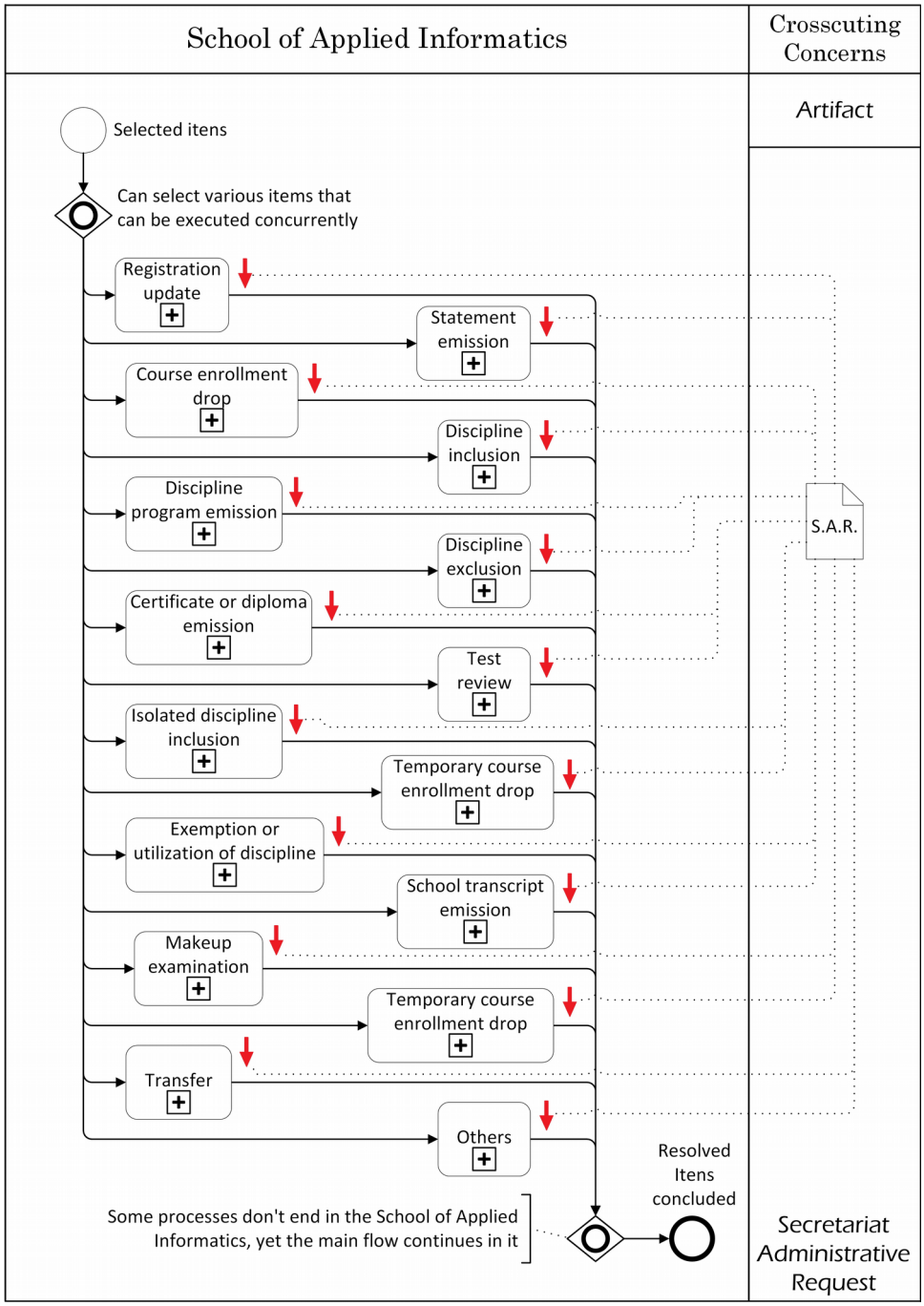

Figure 21: Selected Items Index Process built using AO-BPM.

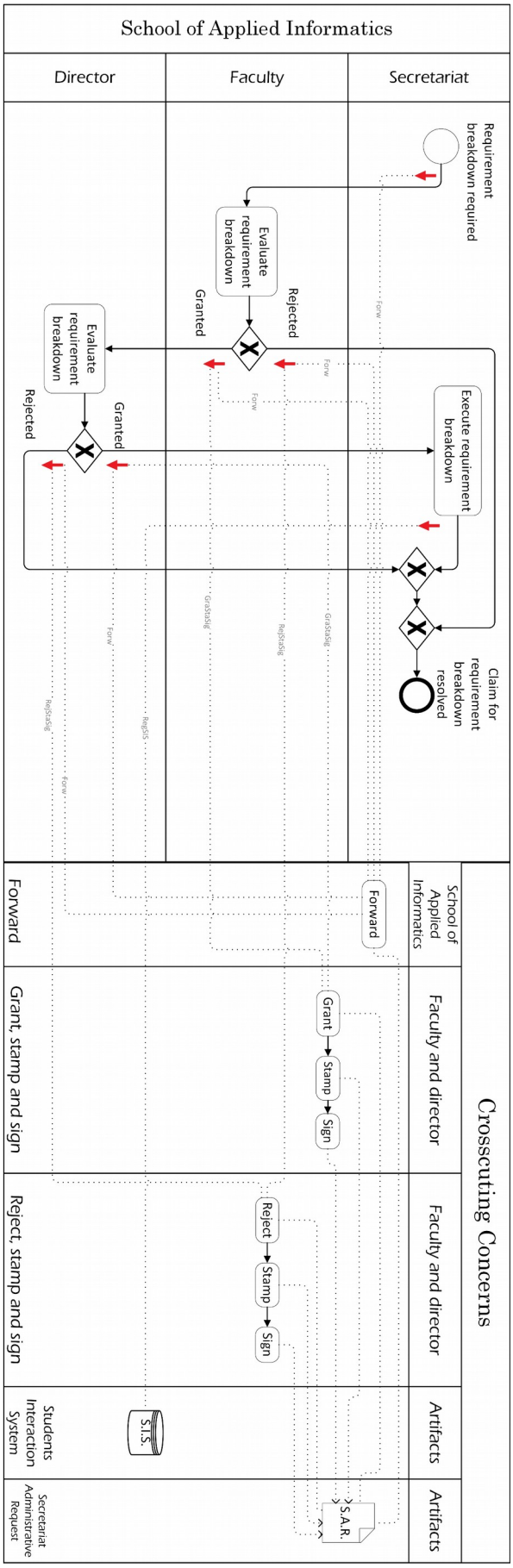

Figure 22: Requirement Breakdown Process built using AO-BPM.

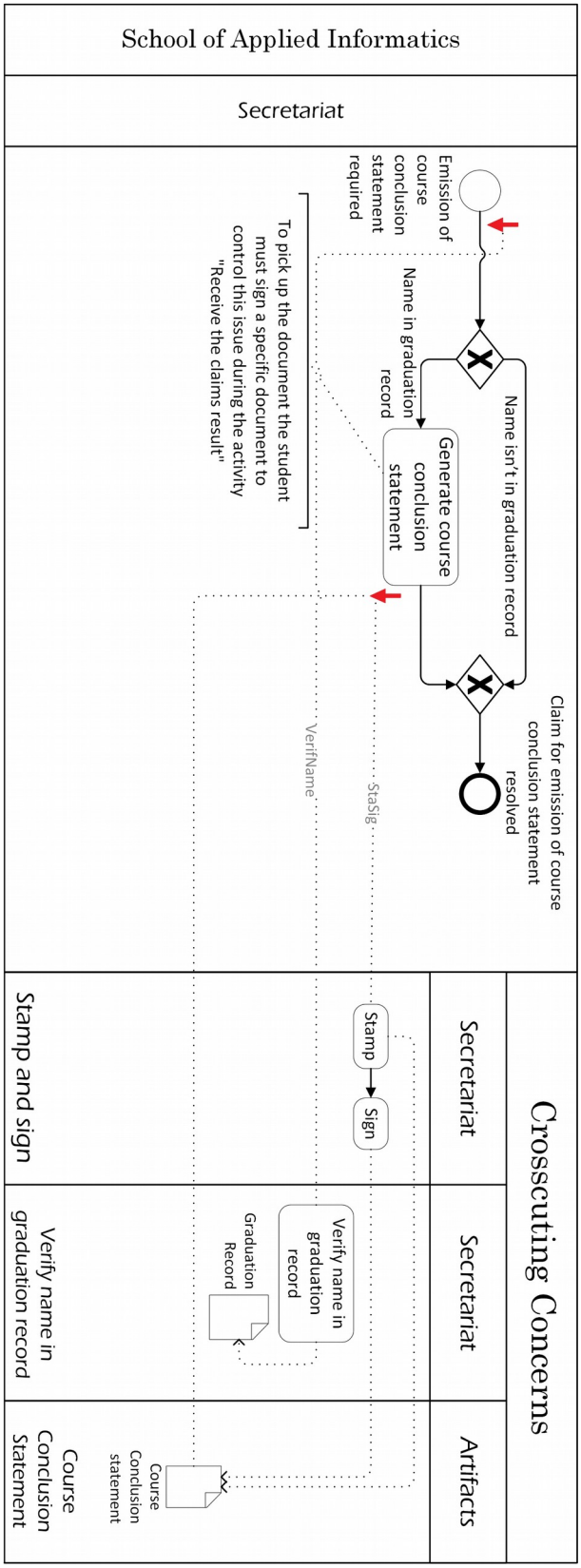

Figure 23: Course Completion Statement Process built using AO-BPM.

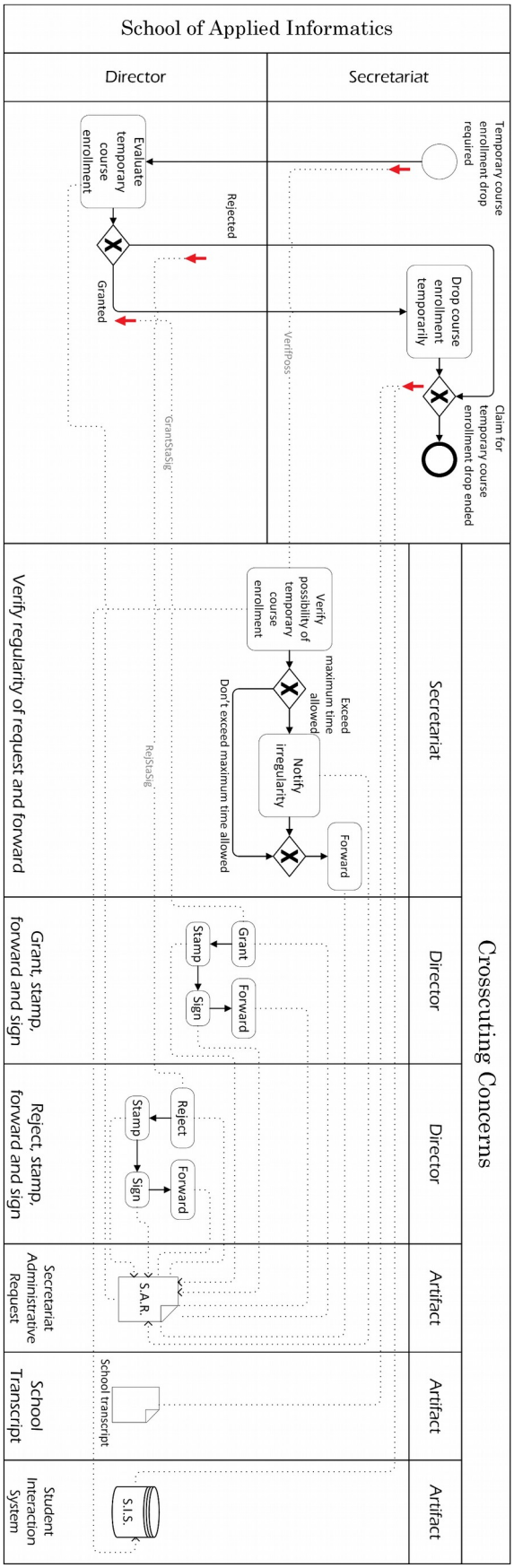

Figure 24: Temporary Course Enrollment Drop Process built using AO-BPM.

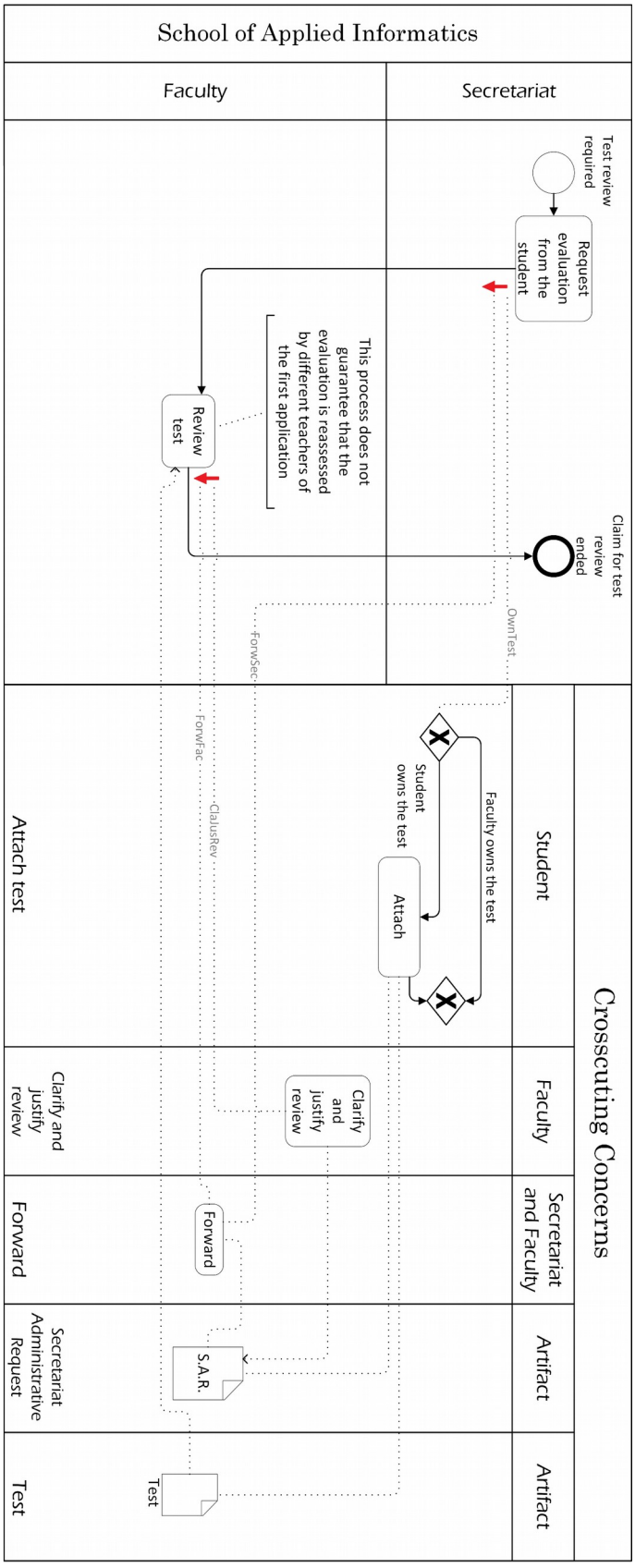

Figure 25: Test Review Process built using AO-BPM.

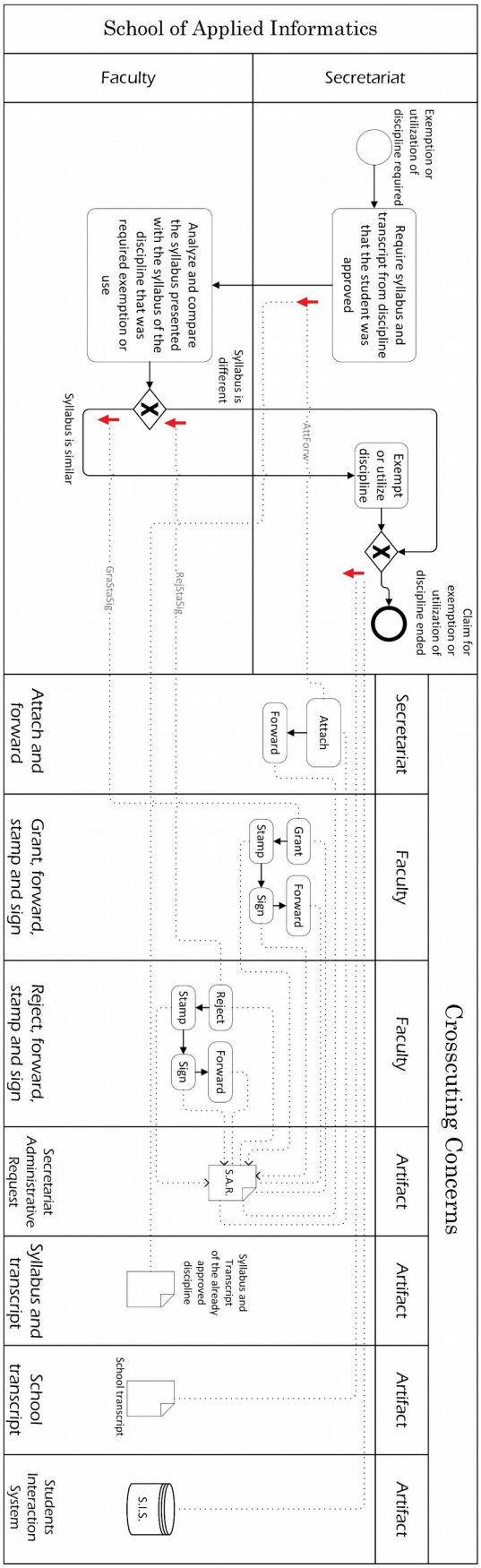

Figure 26: Exemption or Utilization of Discipline Process built using AO-BPM.

Some characteristics about an operationalization of AO-BPM in a business process portfolio can be observed, not just one (SANTOS *et al.*, 2011) (CAPPELLI *et al.*, 2010), and with different levels of aspect-driven efficiency, the following conclusions are made:

- The crescent number of crosscutting relationships hurts the easiness to read for two main reasons. The first is the split-attention effect, but not only for track the aspect but to read the pointcut language to understand and order the same advices related to the same pointcut. The second is the difficult related to directly track and locate the advice itself, in a process model already harmed by compositeness the only task of eye-track the crosscutting relationship from the pointcut to the advice is a challenge alone. A reader may spend more time tracking, following crosscutting relationships and comprehending the pointcut language than absorbing the modeled information.
- The concrete syntax proposed by AO-BPM to display encapsulated items in the cross-cutting concern pool with a lane for each one hurt the simplicity in a scenario with a horizontally long cross-cutting concern pool. Greater the distance between the aspect and the pointcut harder the eye-tracking between them.
- Crosscutting relationship flow naming is not trivial. Two advices having lightly differences, in a composite scenario, can lead to overthink about labeling the flow to match the correspondent information pool in the pointcut language.
- The separation of resources is scarcely clarified; cannot know exactly who performs the encapsulated elements in Figure 22, just get this information when match the advice with the location of the pointcut in the model. In this setting, if the model has many resources associated with the same aspect the problem gets worse, if in Figure 22 the model presented a lane for each actor performing the “Forward” task, the model would be so great that would be no gain in the simplicity and easiness using that aspect-driven approach, the advice tracking and readability would be so harmed and the size of the crosscutting concern pool so big that the best solution would be using a fully flattened solution and not the aspect one.
- AO-BPM cannot, in the actual version and in this setting, syntactically operationalize something like a global task or global process solution (as proposed for AO-BPM 2.0 in Chapter 4.4.1). With the limitation of the crosscutting relationships grounded and the crosscutting concern pool growing from the right side of the core process pool, there is no solution to implement a crosscutting process pool of repeated elements in

many crosscutting process pools, limiting the scalability of the aspect-oriented solution.

Based on this analysis this work does not compare the models in AO-BPM with BPMN and AO-BPM 2.0. AO-BPM for having scalability, modularity, simplicity and easiness problems those need to be solved, principally in a large scenario use.

5.3 AO-BPM 2.0 operationalization

This Chapter presents the models built using AO-BPM 2.0, followed by the same models in BPMN, and the value-adding analysis of the activities on them.

5.4 General aspects

Using the proposed operationalization in Chapter 4.4.1 and observing the models present in Chapter 5.2, certain aspects repeat on several models, including many resources responsible for the implementation of certain aspects, exposing the operation in Chapter 4.4.2. Consequently, a pool of general or crosscutting aspects was created serving as an aspect index in the model collection of the proposed scenario.

Note the repetition of the evaluation and referral activities of the application, the atomic task of stamping in most sub processes contained in the main processes. The potential objects eligible as aspects identified were: SAR form, the academic transcript and the Students Interaction System (SIS) database. Even aspects that interact with others, e.g. aspect 7 (Figure 27), in a behavior allowed in Tavares and Marinho (2014), aspects inside aspects.

Constantly repeated the "Director", "Faculty" and "Support Staff" resources performing specific aspects during the flow, generalizing them as aspects they may be invoked using their respective categorizer’s suffixes.

The next sub-chapters present all the process models in this case study and the aspect-oriented modeling operationalization with AO-BPM 2.0.

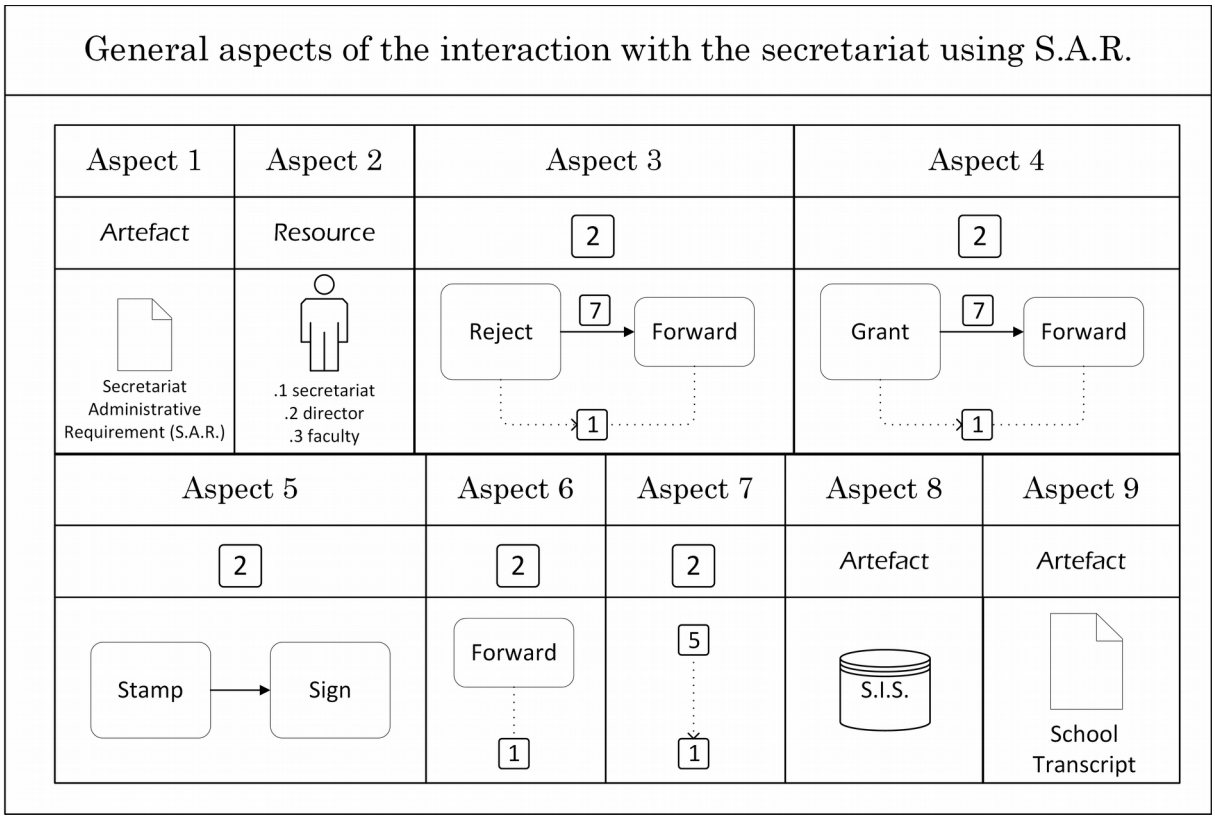

Figure 27: General aspects pool of the interaction with the secretariat using S.A.R.

5.4.1 Interaction with the Secretariat

This process represents the claimant's interaction with the secretariat since it requires a procedure that needs the SAR to receive the results. The participation of the secretariat is circumstantial and it is only about handling the document that provides the trigger to the sub processes tied to it, its activities, as soft-goal were classified as aspects and arranged on the embedded pool, since they are not shared with any further process in the scenario. As this process has a high level of particularity only one aspect uses the general aspects pool, embedding the remainder.

The essentially functional objects to the objective end of the process are not aspects, as noted earlier. It was found that the interaction of the claimant, the student and customer, occurs mainly to select the procedures one needs, the procedures themselves and the request response made earlier; being the other aspects a support to this macro process.

The models built using BPMN and AO-BPM 2.0 are, respectively, represented in Figures 28 and 29.

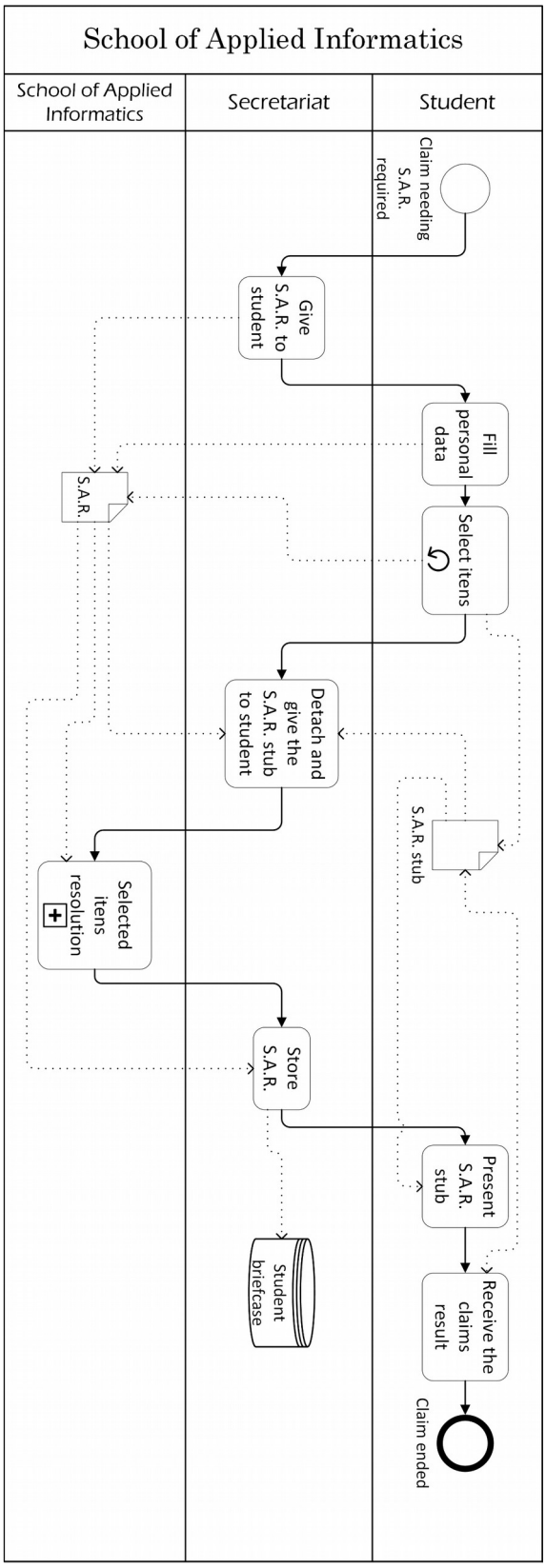

Figure 28: Interaction with the Secretariat Process built using BPMN.

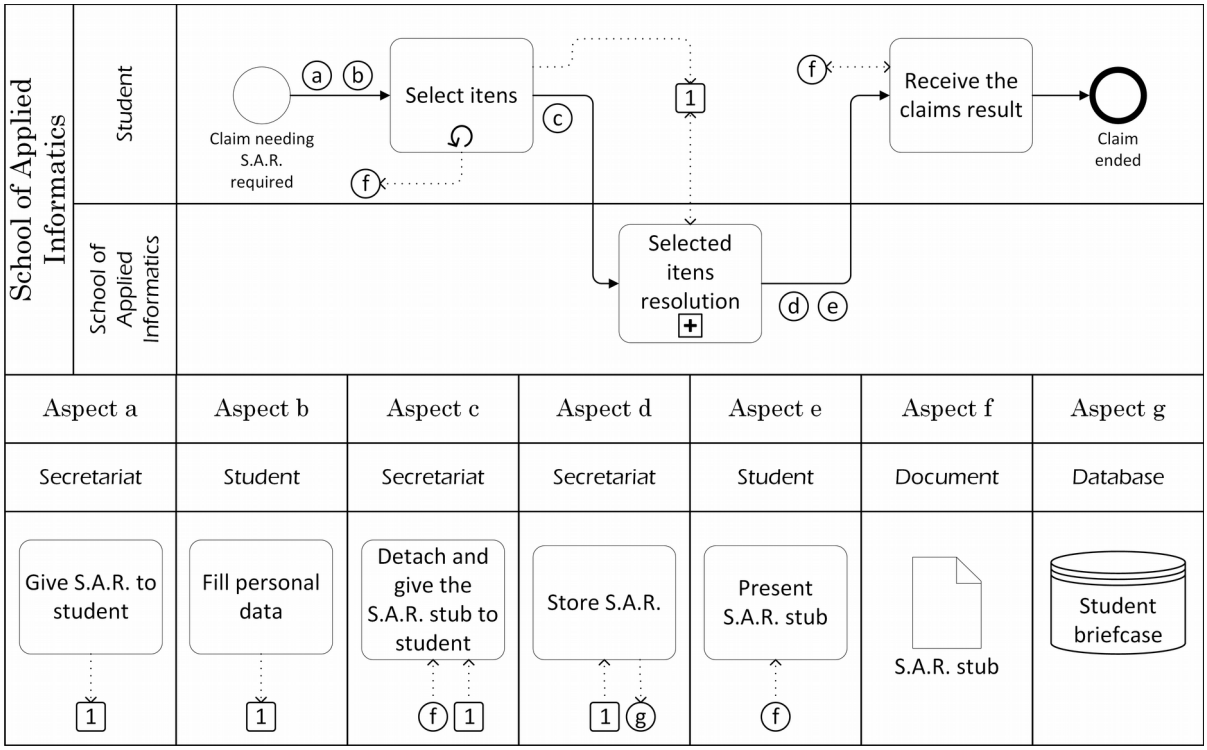

Figure 29: Interaction with the Secretariat Process built using AO-BPM 2. 0..

5.4.2 Selected Items Index

This process represents each case ordered in SAR form, possibly selecting multiple items and various processes may occur concurrently. For example, if required two assertions, three requirements break and two discipline exemption processes in the same form they can occur simultaneously, pegging each result to the SAR individually.

In this process the abstraction obtained by sub leveling of process is already clear, if items are created, deleted or modified in the SAR the upper level will remain unchanged, only the form must be modified in view of the changes. Thus, it acts as a layer of integrity among the processes, in fact, carried out and the claimant's interaction with the secretariat.

It has only one aspect, the SAR object, since its particularity is specific due to its operation as index for mapping other sub processes, all other objects present are essential to the process goal.

This sub process is an expanded and the direct son of the macro process "Interaction with the secretariat".

The models built using BPMN and AO-BPM 2.0 are, respectively, represented in Figures 30 and 31.

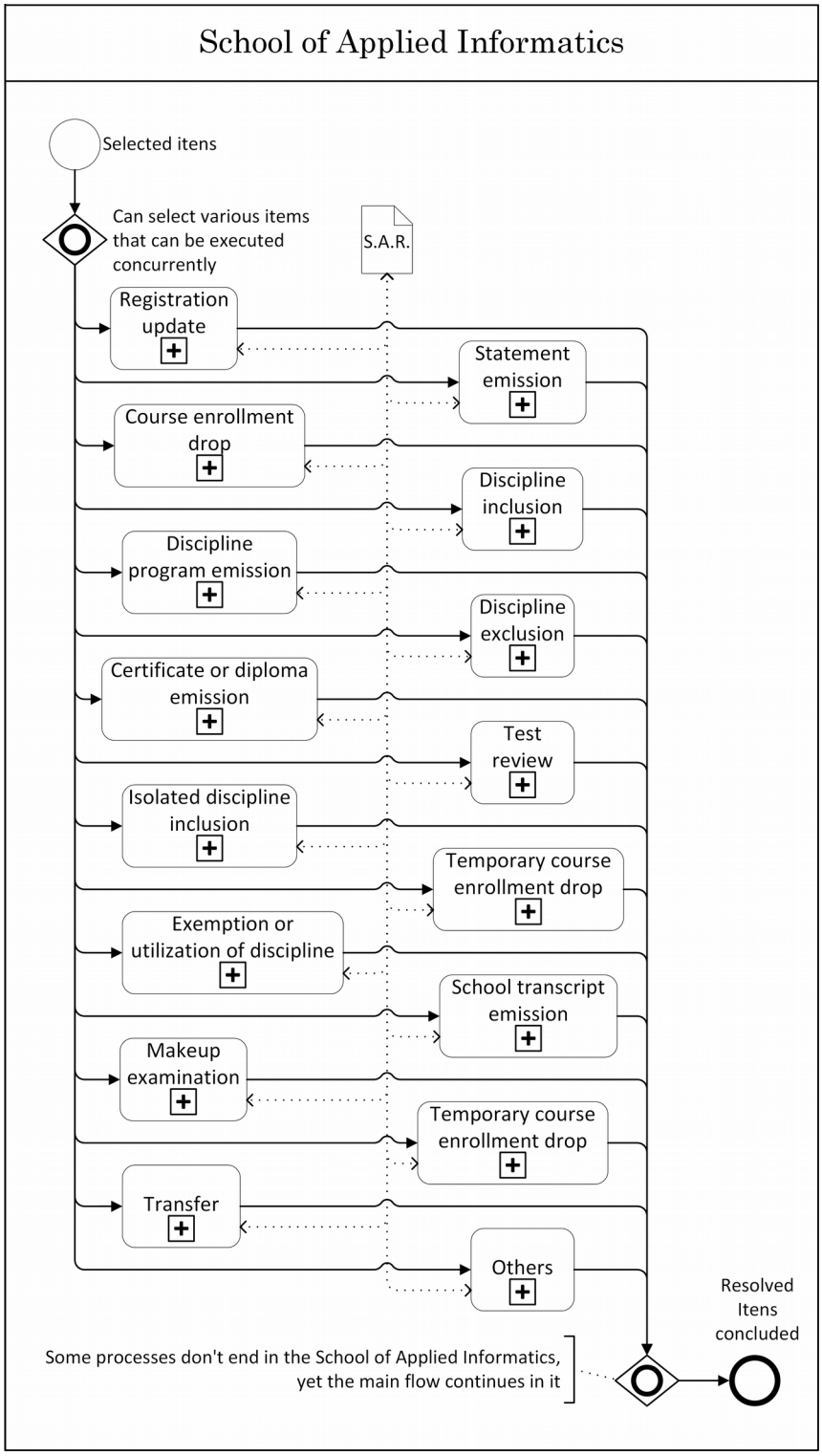

Figure 30: Selected Items Index Process built using BPMN.

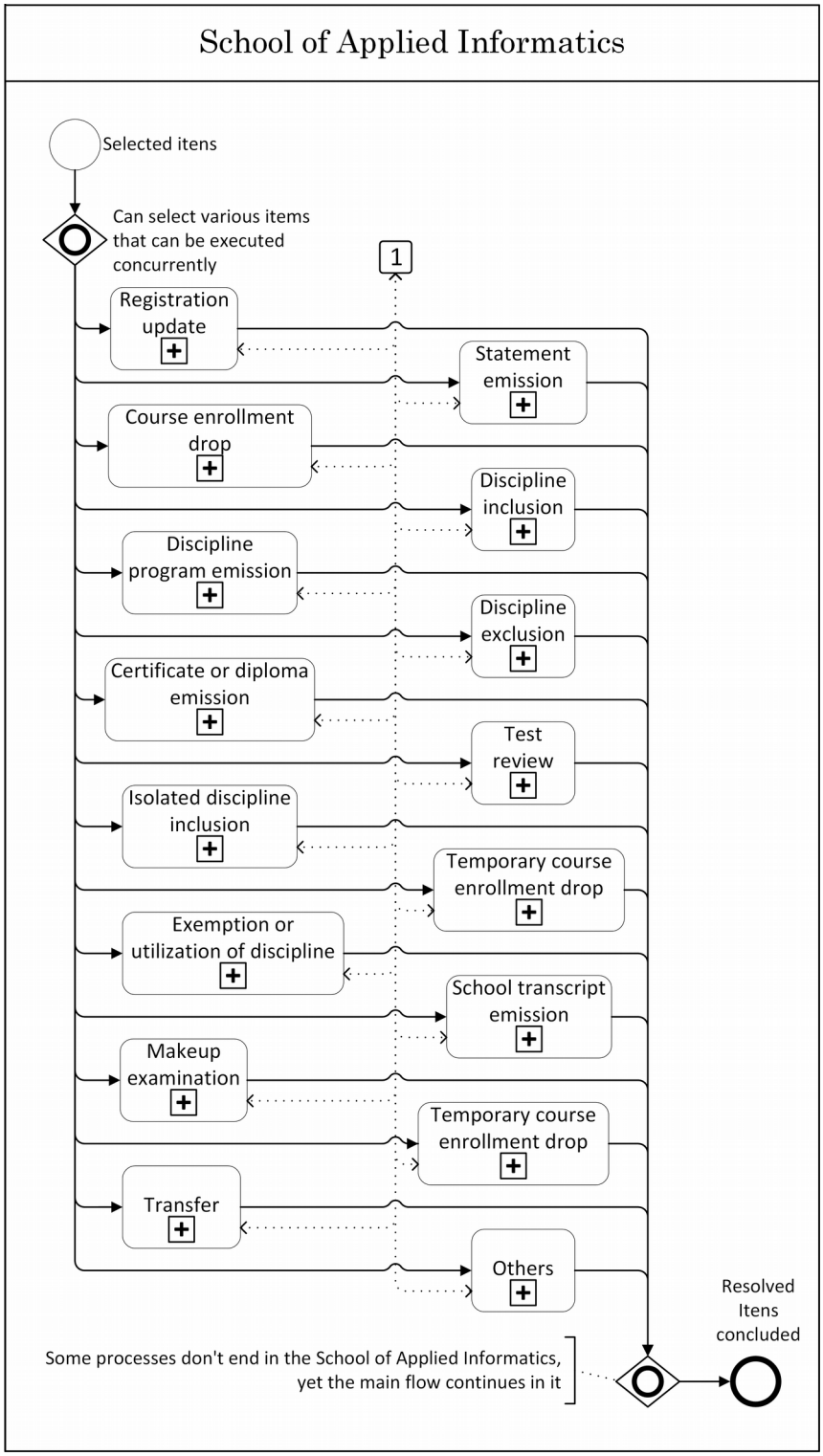

Figure 31: Selected Items Index Process built using AO-BPM 2. 0..

5.4.3 Selected items

This Chapter groups the claimant's selected items, their models of individual processes. Fifteen sub processes identified and one sub process that groups other less required and relevant, "Others". Presenting five sub processes here, where the aspect-oriented modeling is clearer in them.

At this level, where each sub process is hierarchically brother, the aspects are clearer and identifiable, the occurrence of them in each effective process. Items in the general aspects pool are widely present here, for example, a process importing all its aspects (sub-chapter 5.4.3.1) to processes which there will be at least one importation (sub-chapter 5.4.3.2).

Each sub process below is an expanded and direct son of "Selected items index".

5.4.3.1 Requirements breakdown

This process represents the claimant's application to attend a discipline that one does not meet the requirements, "breaking" them with guarantee of direction and respective faculty member, allowing application in it. This item is contained in the set of process "Other" sub processes.

The scope of that process is the breakdown of requirements, relevant to that: the analysis and the opinion of each evaluator of the application and consolidation of it in the system to complete the intention; the other objects supports indirectly these functional activities being aspects or soft-goals (transport, authentication, bureaucracy and data objects).

In this instance, the general aspects pool instrumentation (Chapter 4.4.1) and redundant resources (Chapter 4.4.2) are clear, there is no embedded aspect and the resources repeat several times. Presenting the biggest difference between traditional modeling and aspect-oriented, among the models presented in this work.

The models built using BPMN and AO-BPM 2.0 are, respectively, represented in Figures 32 and 33.

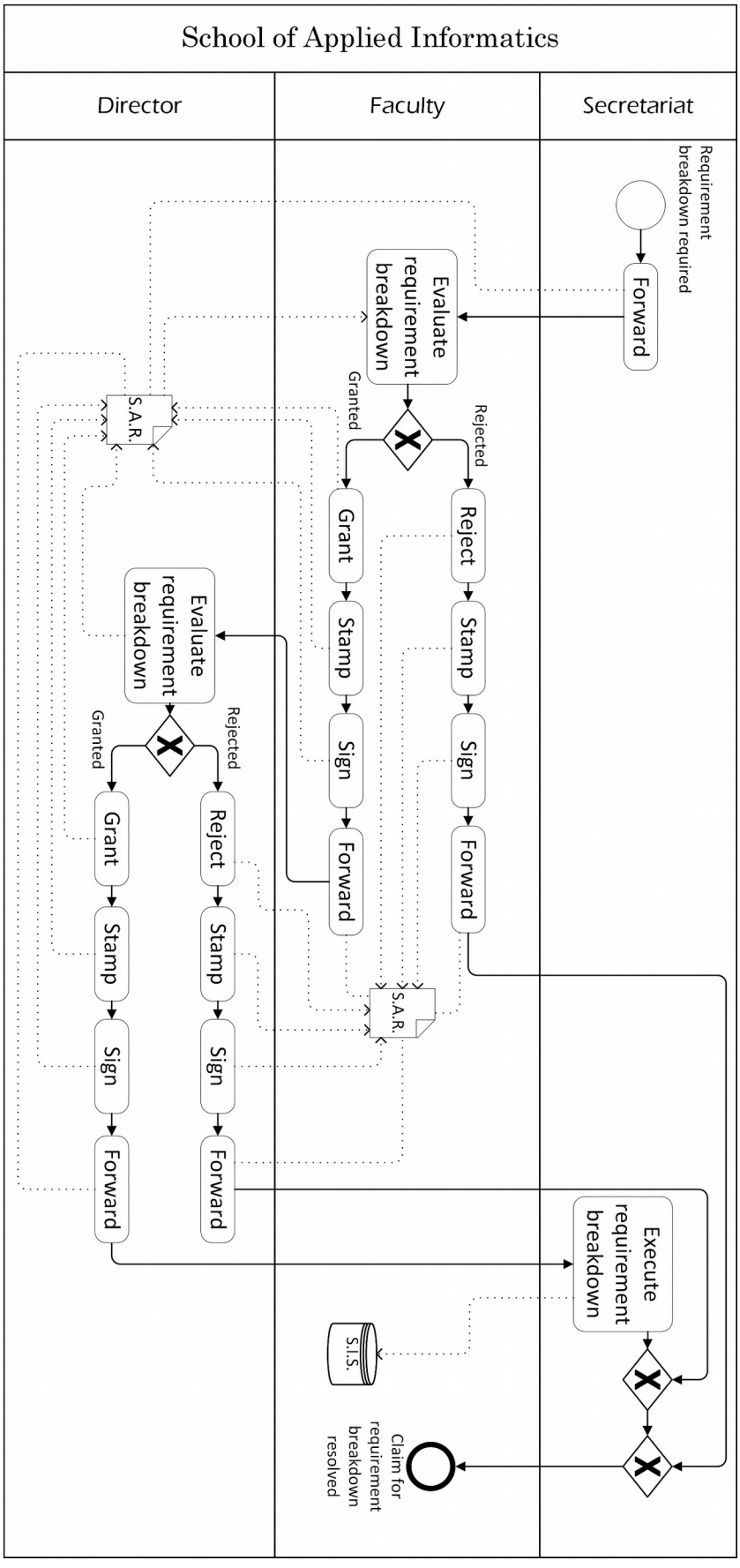

Figure 32: Requirement Breakdown Process built using BPMN.

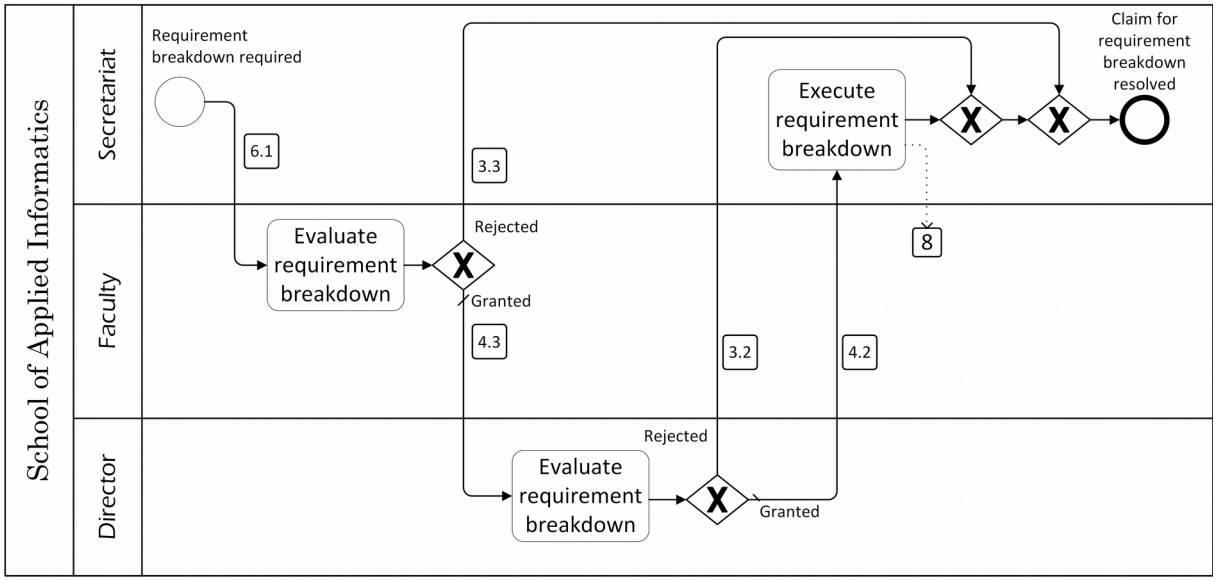

Figure 33: Requirement Breakdown Process built using AO-BPM 2. 0..

5.4.3.2 Course completion statement

This process represents the claimant's application to a formal statement emission confirming his course completion, respecting the necessary time to have the diploma done and obtaining the same. This item is contained in "statement emission"; from the moment that the applicant notifies that the desired type of statement is "course completion".

The scope of the process is the generation of a declaration confirming the conclusion of the course by the claimant. There is a verification activity to ensure that the claimant indeed already completed the course, one`s name shall be on a list with the graduate’s names. The verification of these minutes influences the flow of the process since it can only occur if the claimant's name appears in this document, however it is only one "trigger" activity and does not influence the generation of the same, being a safety aspect.

The models built using BPMN and AO-BPM 2.0 are, respectively, represented in Figures 34 and 35.

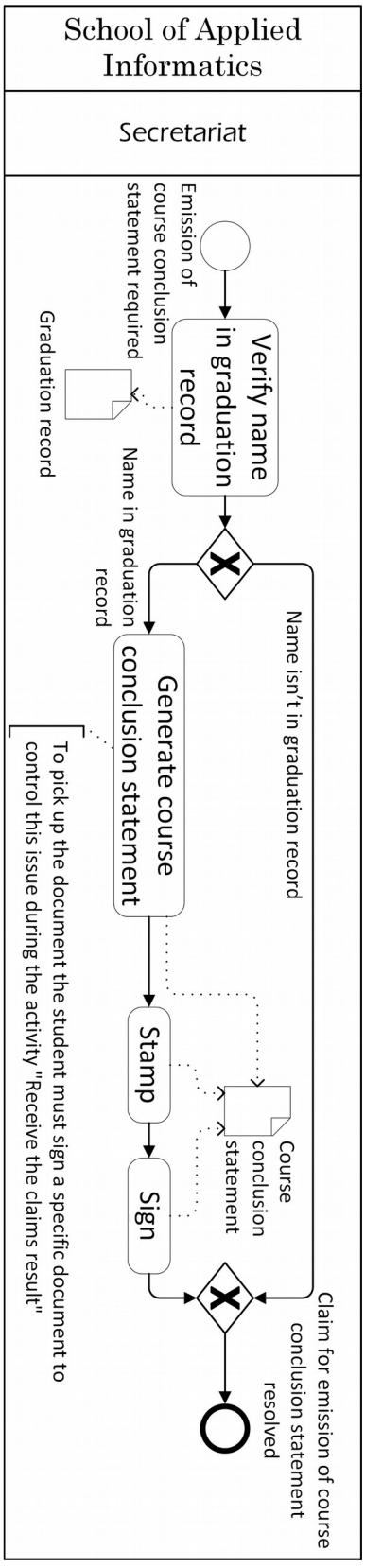

Figure 34: Course Completion Statement Process built using BPMN.

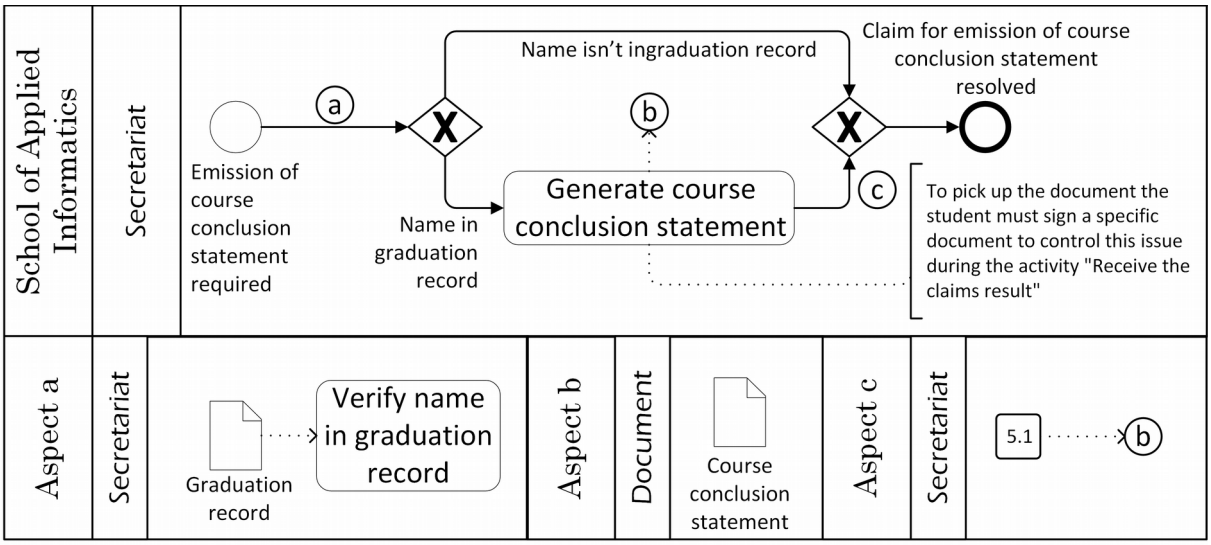

Figure 35: Course Completion Statement Process built using AO-BPM 2. 0..

5.4.3.3 Temporary course enrollment drop

This process represents the claimant's application to drop the course enrollment for a while. This item has sub own process called "Temporary course enrollment drop".

The scope of the process is the course enrollment dropped temporary for the claimant, authorized by the director. An organization rule imposes a maximum time of dropping, different from the "Course completion statement" process the director needs to control this application; one more verification aspect exists to notify the irregularity in the SAR and not forwarding it to the end. Important for this process is the request evaluation and the dropping performed, other tasks support these.

The models built using BPMN and AO-BPM 2.0 are, respectively, represented in Figures 36 and 37.

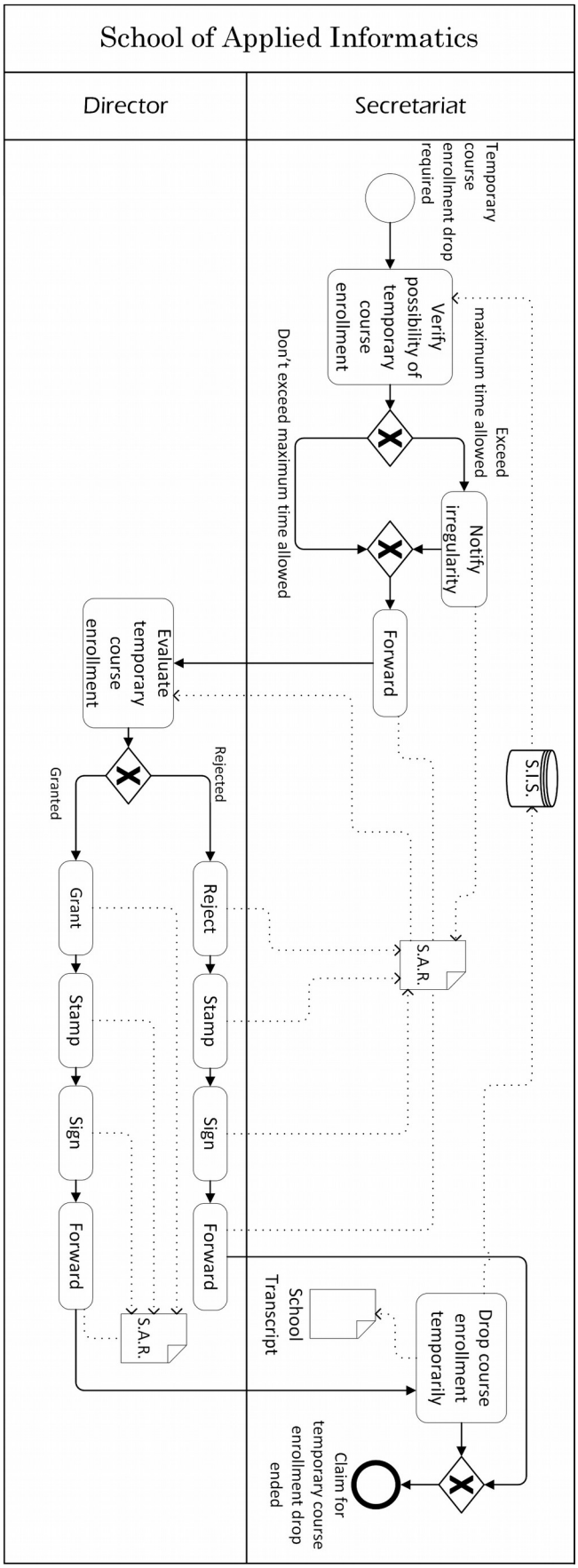

Figure 36: Temporary Course Enrollment Drop Process built using BPMN.

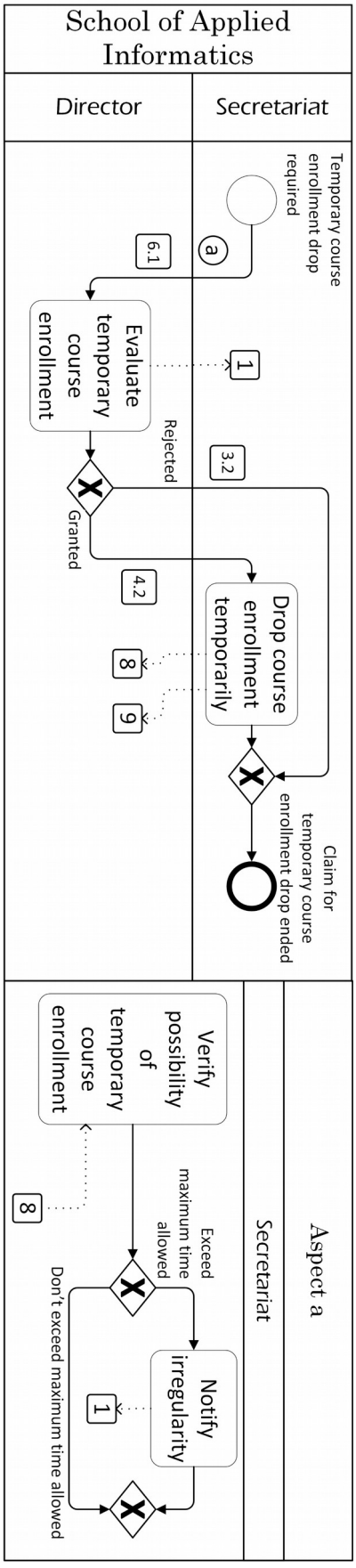

Figure 37: Temporary Course Enrollment Drop Process built using AO-BPM 2. 0..

5.4.3.4 Test review

This process represents the claimant's application for a review of a test one did. This item has sub own process called "Test review".

The scope of the process is the review of a test already carried out; the institution or the applicant can hold it. Consider relevant only the test revision itself. The first activity is essential to get the operationalized document used by the main activity of the process, the review. The clarification and justification of it, after the review, simply overlooked if there is no change in the evaluation, becoming one aspect.

The models built using BPMN and AO-BPM 2.0 are, respectively, represented in Figures 38 and 39.

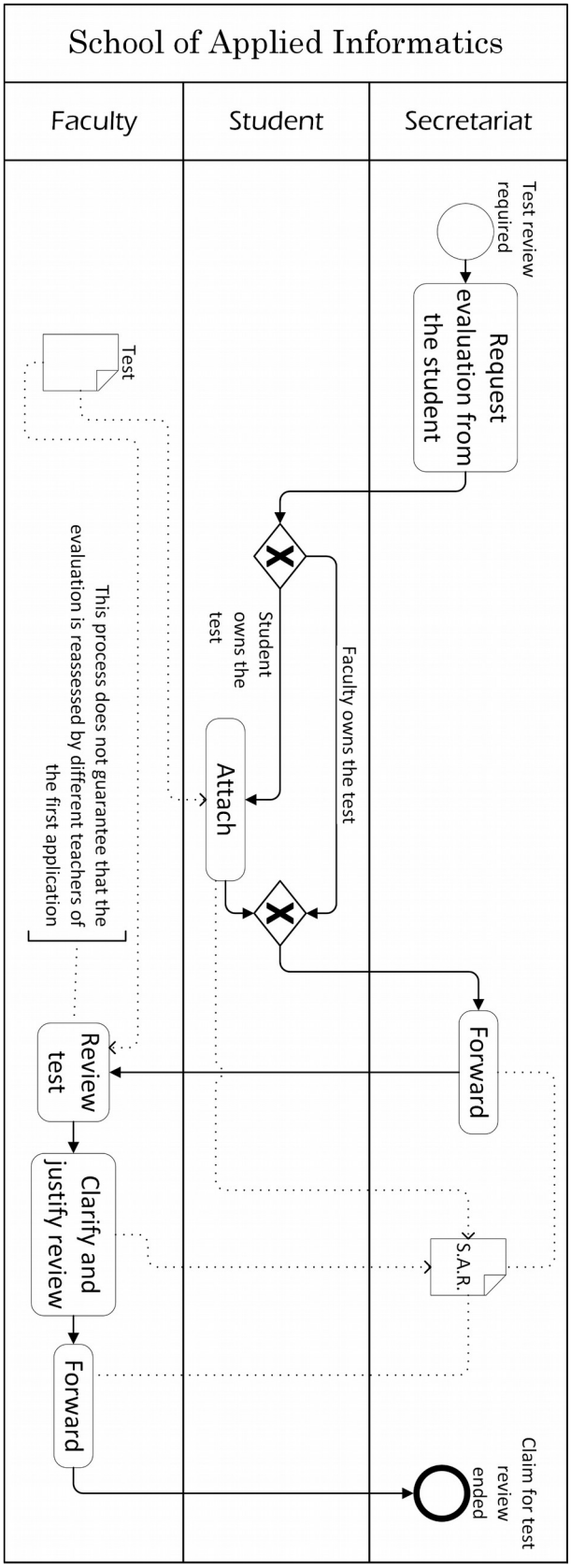

Figure 38: Test Review Process built using BPMN.

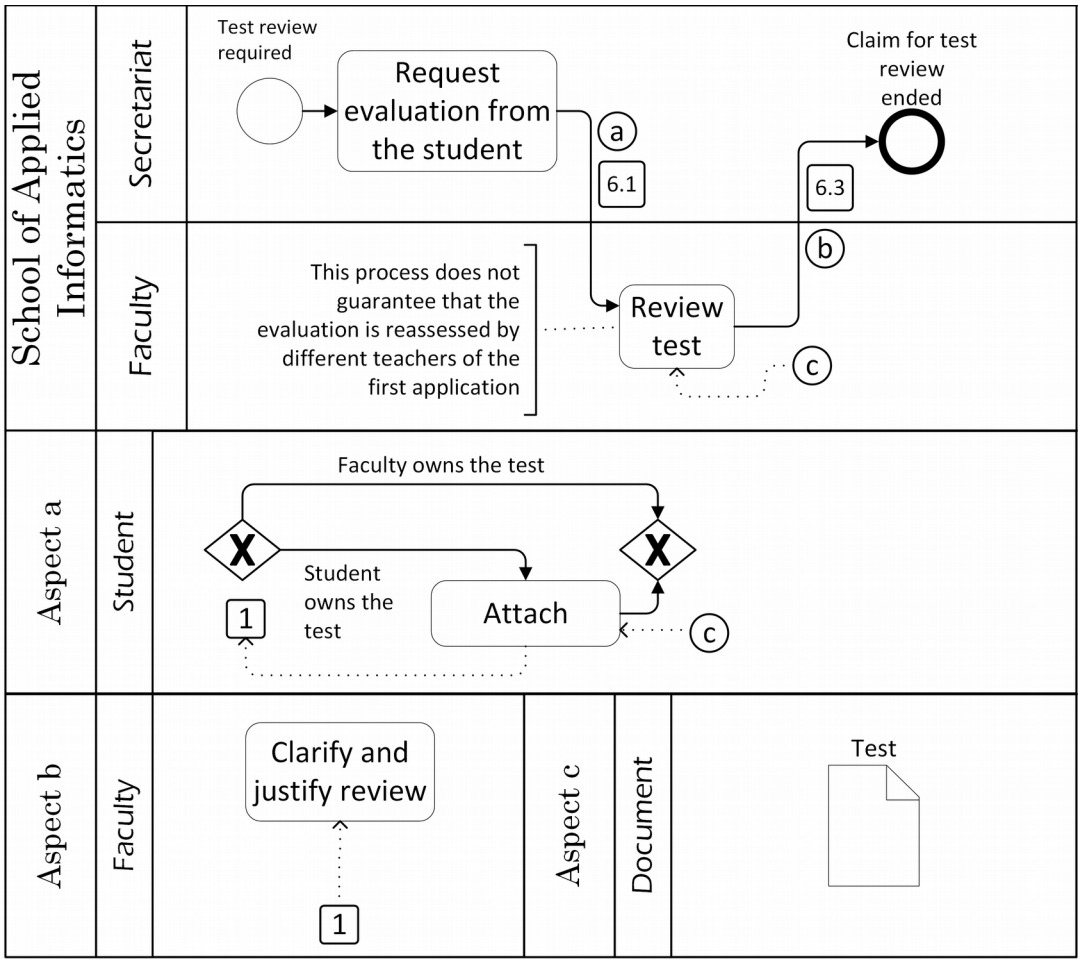

Figure 39: Test Review Process built using AO-BPM 2. 0..

5.4.3.5 Exemption or utilization of discipline

This process represents the claimant's application intending to exempt or utilize approved previously disciplines. This item has sub own process called "Exemption or utilization of discipline." When the applicant has been approved in another course discipline of UNIRIO, he uses the same, while disciplines external to UNIRIO are exempt.

The scope of the process is the exemption or utilization of a previously approved discipline the student completed in another course. The first activity is essential to get the document exploited by the main activity of the process, the examination of the application. Relevance takes place in the examination of the application, which will culminate in the discipline exempt/utilization or rejection culminating in the end of the process.

The models built using BPMN and AO-BPM 2.0 are, respectively, represented in Figures 40 and 41.

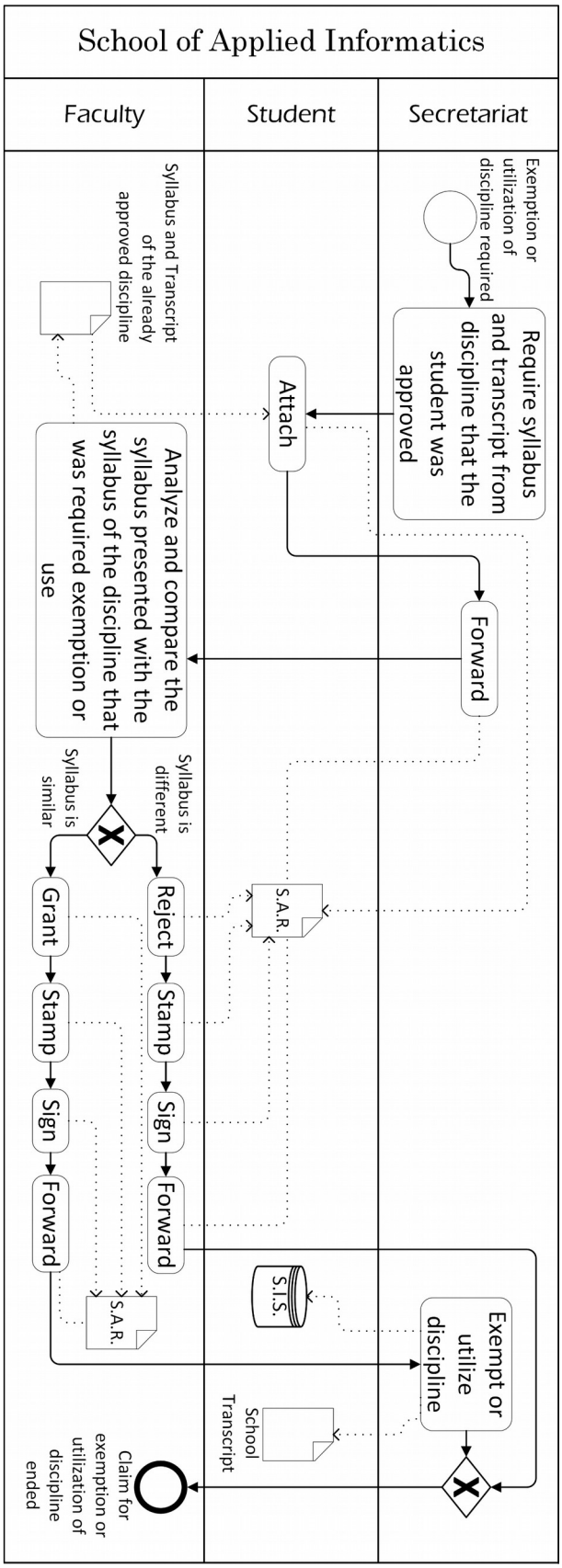

Figure 40: Exemption or Utilization of Discipline Process built using BPMN.

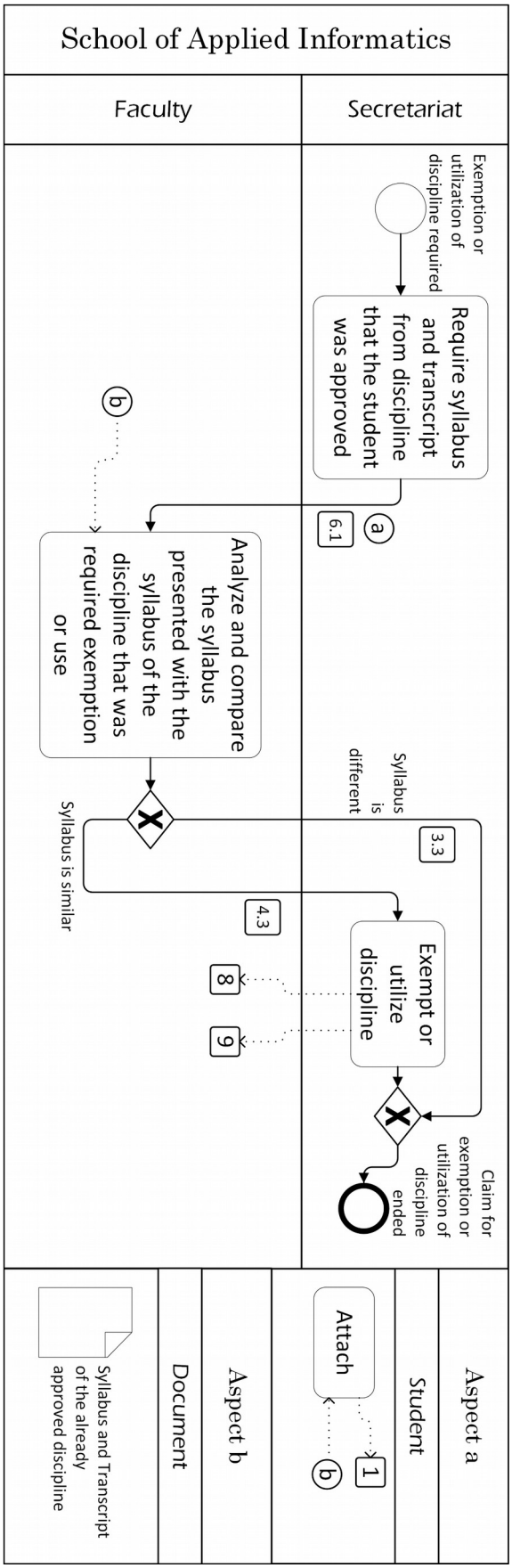

Figure 41: Exemption or Utilization of Discipline Process built using AO-BPM 2. 0..

6 Discussion

The scenario of the case study was propitious for aspect-oriented implementation, given that the process of interaction with the secretariat is not the main purpose of the organization, so much of the data objects and activities become potential aspects.

Process models generally include the activities and tasks not strictly linked to the main scope of the process. Miscellaneous transport activities, validation, verification, transparency, maintenance, etc. that support and ensure its executions combined with low relevance activities, i.e. soft-goals, can be segregated without hurting the interpretation of its main functionality. This segmentation raises the level of modularization, which in the traditional modeling was limited to sub processes and subjective atomicity of activities and aspects modeling also encompasses non-functional activities, soft-goals and certain data objects. The modularization allows updating the data segregated from the main pool without updating the same, creating a preservation layer between the main pool and the aspects pool. For example, if there is a change in the document authentication procedure by removing the task "Sign" and keeping only "Stamping" you only need to update the authentication aspects rather than updating each authentication procedure contained in each process.

There was a noticeable decrease in the processes models size, helping the simplicity. Inversely proportional, its easiness of comprehension decreased, considering the invocation of the general features and the embedded features to complete the model interpretation from an external repository.

6.1 LOC metric result

Table 2 shows the results of the LOC metric, an adaptation of NOAJS, in the case study. The processes are identified in the table by the acronym of their names, and respectively in the order presented in this work.

Table 2: LOC Metric results of the case study

	IS	SII	RB	CCS	TCED	TR	EUD	Total
BPMN	11	19	27	8	21	10	20	116
AO-BPM 2.0	3	19	7	3	4	2	5	43
AO-BPM 2.0 + pointcuts	11	19	13	6	11	7	12	79
Calls for crosscutting concern pool of general aspects	5	1	6	1	8	4	6	31

The metric used, adapted NOAJS, included elements in the core process: activities, data objects, splits and joins. It dismissed initial and final events, because of the rules of syntax and semantics of AO-BPM 2.0 (Chapter 4.2) the same cannot be encapsulated as aspects, and textual annotations, which only aim to inform and not functionally influence the process.

The reduction of the number of objects in the core process is clear. In the process Requirement Breakdown, the number of elements is reduced by more than 74%, while in the Selected index number is kept, by the process structure. The interaction with the secretariat process can be summed up to 27,2% of their activities if they are considered only the functional activities and necessary for the same, adding artifacts that invoke aspects to demonstrate the activities weakly functional or nonfunctional the gain is zero.

All processes used to process crosscutting concern pool of general aspects. His operation was effective and prevented redundancy aspects of the portfolio. In the process Requirement Breakdown, for example, the gain from this pool is 100%, demonstrating the benefit of this object.

The low-level hierarchy processes (excluding interaction with the secretariat and selected index item) have more redundant aspects and crosscutting concerns to each other and in themselves. The gain from the aspect orientation was higher in them.

In total, there was a 63% reduction in the objects in the case study models. Removing the weakly functional elements or nonfunctional left over only 37% of the elements, which illustrate directly and functionally the objective processes. Counting artifacts that connect the core process in the crosscutting concern pools (pointcuts) reduction is 32%, amount of redundancy eliminated using the aspect orientation.

The aspect orientation needs dedicated metrics. These metrics can get up to assess whether the orientation aspect is favorable or recommended in each scenario. In scenarios

with little redundancy or where there will only shift objects to the crosscutting concern pool the benefits may not correspond with the cost of using this paradigm.

The results of the LOC metric in this case study demonstrate a positive impact on simplicity using aspect orientation. By reducing redundancy, separating concerns, clarifying on functional elements or not, increasing flexibility.

7 Conclusion

The work proposed in the beginning of this undergraduate thesis was complete. Successfully compared the BPMN, AO-BPM and AO-BPM 2.0 and analyzed the use of the last one to build model and handle aspects, soft-goals and non-functional elements.

Contributions, related work, limitations and difficulties and proposal for future work are presented to conclude this work.

7.1 Contributions

The approach proposed shows a different form of process modularization, which showed able to simplify the models and make some elements more flexible and potentially reusable. Through the evaluation performed, it was possible to observe that the notation contributes to decrease redundant aspects spread between processes.

Clarify and gather terms of aspect orientation intended to represent business processes, with a focus on graphic representation.

Analysis and comparison of four aspect oriented modeling notations in the scope of conceptual modeling, discussing some drawbacks and characteristics of it. A detailed analysis of the AO-BPM 2.0 notation, deepening and complementing the work of Tavares and Marinho (2014), aiming also to demonstrate the qualities of this notation as a conceptual graphical representation of business processes.

Operationalization of AO-BPM 2.0 in a real setting, comparing it with two other notations, BPMN and AO-BPM. Demonstration and discussion of its modularity capabilities, from specific metrics, reducing and simplifying the models, eliminating redundancy (embedded and in the portfolio), representing resources as aspects respecting their singularity and uniqueness, and so on.

7.2 Difficulties

The inexistence of a dedicated modeling tools for aspect orientation process models symmetric to BPMN (AO-BPM and AO-BPM 2.0). All the models were drawn on the Microsoft Visio tool and the online drawing tool draw.io.

The difference between i) aspects or soft-goals and ii) actual value adding, business value adding or non-value adding activities strictly relevant for the process scope is very subjective. A direct and objective guideline or “cake receipt” to discover and elicit aspects and soft-goals in process models does not exist, because they are related not just with the enterprise value chain as to the mission, vision and values of the organization. If one of the values of determined organization is, e.g. transparency, the raisin question will be: “Can we pick the activities related to transparency and turn them in aspects or soft-goals?”. In that case a deep business analysis, i.e. the goal of the information representation itself, needs to be done. This work cannot provide that analysis, just the recommendations in Chapter 3.

The main difficulty of this work was the scarcely available material about the aspect oriented paradigm and aspect oriented business process modeling, demonstrating partly the need for more work to be done and apparently, some disinterest from the academy by the aspect orientation paradigm in business process.

7.3 Future Work

As noted in Chapters 3 and 4 AO-BPM 2.0 has some loose ends, the future works presented here aim to tie this issue.

The absence tool or functionality for modeling weakens the use of notation (JALALI, WOHED and OUYANG, 2012), the development of an effectively operational AO-BPM 2.0 modeler solves this problem with the features respecting the syntactic and semantic boundaries of notation, directly linking the pointcut with the aspects, providing sequencing and automatic organization of contoured elements (e.g. when one of them is modified or deleted they reorganize automatically), etc.

Conversion methods (MOULINE and LYAZIDI, 2013) of AO-BPM 2.0 to formal languages (VAN DER AALST, 2013), e.g. Petri Net, allowing objective analyzability.

A further deepening in the study of communication between the core process and crosscutting concern process is required in AO-BPM 2.0, this occurs by weaving. Other notations or languages already have their weaving as AO4BPMN (WITTERBORG *et al.*,

2014) and AOBPMN (JALALI *et al.*, 2013), the study of static or dynamic weaving in AO-BPM 2.0 is required, even for analysis of performance. The semantics of the modeling is different from the implementation of semantics (MENDLING, REIJERS e VAN DER AALST, 2010).

8 References

BALDAM, R.; VALLE, R.; ROZENFELD, H. **Gerenciamento de Processos de Negócios – BPM:** uma referência para implantação prática. Rio de Janeiro: Elsevier, 2014.

BELLMAN, R. The theory of dynamic programming. **Bulletin of the American Mathematical Society**, 60, 1954.

BRANDÃO, B. **Aspects Identification In Business Process Through Process Mining.** Master's Thesis, Federal University of the State of Rio de Janeiro. Rio de janeiro. 2015.

BRANDÃO, B.; SANTORO, F. M.; AZEVEDO, L. **Towards Aspects Identification in Business Process Through Process Mining**. XI Brazilian Symposium on Information System. Online: AIS Electronic Library. 2015.

BRICHAU, J. et al. **Aspect-Oriented Software Development:** an Introduction. Online: Wiley Encyclopedia of Computer Science and Engineering. 2008.

BYTHEWAY, A. **Investing in Information. The Information Management Body of Knowledge**. Berlin: Springer-Verlag, 2014.

CAPPELLI, C. et al. Reflections on the modularity of business process models. The case for introducing the aspect-oriented paradigm. **Business Process Management Journal**, Bradford, v. 16, n. 4, 2010.

CAPPELLI, C. **Uma Abordagem Para Transparência Em Processos Organizacionais Utilizando Aspectos**. Doctorate's Thesis, Pontifical Catholic University of Rio de Janeiro. Rio de Janeiro. 2009.

CARDOSO, J. et al. A Discourse on Complexity of Process Models. In: EDER, J.; DUSTDAR, S. **Business Process Management Workshops**. Berlin: Springer-Verlag, v. 4103, 2006.

CARVALHO, L. P.; SANTORO, F. M.; CAPPELLI, C. **Representação e Transparência dos Processos da Secretaria da Escola de Informática Aplicada da UNIRIO**. Online: RelateDIA. 2015.

CARVALHO, P. ÉPOCA NEGÓCIOS. **Como a Smart Fit turbinou a receita do grupo Bio Ritmo**, 2014. Available in: <http://epocanegocios.globo.com/Informacao/Resultados/noticia/2014/10/como-smart-fit-turbinou-receita-do-grupo-bio-ritmo.html>. Accessed: 15 nov. 2016.

CHARFI, A.; HEIKO, M.; MEZINI, M. **Aspect-Oriented Business Process Modeling with AO4BPMN**. Berlin: Springer-Verlag. 2010.

CHARFI, A.; MEZINI, M. **Aspect-Oriented Web Service Composition with AO4BPEL**. Berlin: Springer-Verlag. 2004.

DEBEVOISE, T.; TAYLOR, J. **The MicroGuide to Process and Decision Modeling in BPMN/DMN:** Building More Effective Processes by Integrating Process Modeling with Decision Modeling. Amazon: CreateSpace, CreateSpace.

DEVILLERS, M. **Business Process Modeling as a Means to Bridge The Business: IT Divide**. Master's thesis. Radboud Universiteit. Amsterdam. 2011. (156IK).

DUMAS, M. et al. **Fundamentals of Business Process Management**. New York: Springer-Verlag, 2013.

DUMAS, M.; PFAHL, D. Modeling Software Processes Using BPMN: When and When Not? In: KUHRMANN, M., et al. **Managing Software Process Evolution**. 1. ed. Berlin: Springer-Verlag, 2016.

FACEBOOK. **Add Facebook Login to Your App or Website**, 2016. Available in: <https://developers.facebook.com/docs/facebook-login/>. Accessed: 15 nov. 2016.

FISKE, J. **Introduction To Communication Studies**. 3. ed. London: Routledge, 2010.

FRANZ, P.; KIRCHMER, M.; ROSEMANN, M. **Value-Driven Business Process Management:** Impact and Benefits. Queensland: Accenture. 2012.

FUTURELEARN. **About FutureLearn**, 2016. Available in: <https://www.futurelearn.com/about-futurelearn>. Accessed: 15 nov. 2016.

JABLONSKI, S. et al. **Enabling Flexible Execution of Business Processes**. Berlin: Springer-Verlag. 2008.

JALALI, A. **Foundation of Aspect Oriented Business Process Management.** Master’s Thesis, Stockholm University. Stockholm. 2011.

JALALI, A. et al. Dynamic Weaving in Aspect Oriented Business Process Management. In: MEERSMAN, R., et al. **On the Move to Meaningful Internet Systems:** OTM 2013 Conferences. Berlin: Springer-Verlag, v. 8185, 2013.

JALALI, A. et al. Supporting aspect orientation in business process management. **Software & Systems Modeling**, v. 2015, p. 1-23, 2015.

JALALI, A.; WOHED, P.; OUYANG, C. Aspect Oriented Business Process Modelling with Precedence. In: MENDLING, J.; WEIDLICH, M. **Business Process Model and Notation:** 4th International Workshop Proceedings. Austria: Springer-Verlag, v. 125, 2012.

JOSE, H. et al. **Providing semantics to implement aspects**. Berlin: Springer-Verlag. 2016.

KICZALES, G. et al. **Aspect-oriented Programming**. European Conference on Object-Oriented Programming (ECOOP). Berlin: Springer-Verlag. 1997.

LA ROSA, M. et al. Managing Process Model Complexity via Abstract Syntax Modifications. **IEEE Transactions on Industrial Informatics**, v. 7, n. 4, May 2011.

LA ROSA, M. et al. Managing Process Model Complexity via Concrete Syntax Modifications. **IEEE Transactions on Industrial Informatics**, v. 7, n. 2, May 2011.

LAUDON, K.; LAUDON, J. **Essentials of Management Information Systems**. 10. ed. New Jersey: Pearson, 2013.

LIN, F.-R.; YANG, M.-C. P. Y.-H. A generic structure for business process modeling. **Business Process Management Journal**, Bradford, v. 8, 2002.

MELCHER, J. **Process Measurement in Business Process Management:** Theoretical Framework and Analysis of Several Aspects. Karlsruhe: KIT Scientific Publishing, 2012.

MENDLING, J. **Metrics for Process Models:** Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. Berlin: Springer-Verlag, v. 6, 2008.

MENDLING, J.; REIJERS, H. A. **Modularity in Process Models:** Review and Effects in Business Process Management. Berlin: Springer-Verlag. 2008.

MENDLING, J.; REIJERS, H. A.; VAN DER AALST, W. Seven process modeling guidelines (7PMG). **Journal Information and Software Technology**, Massachussets, v. 52, n. 2, February 2010.

MOULINE, S.; LYAZIDI, A. **Formal Verification of BPMN Models using Petri Nets**. 1st International Workshop on Models and Algorithms for Reliable and Open Computing. Morocco: [s.n.]. 2013.

OLIVEIRA, R. FOLHA DE SÃO PAULO. **Smart Fit é a academia mais citada entre paulistanos**, 2015. Available in:

<http://www1.folha.uol.com.br/saopaulo/2015/05/1633852-smart-fit-e-a-academia-mais-citada-entre-paulistanos.shtml>. Accessed: 15 nov. 2016.

OLIVEIRA, S. et al. **Análise e Melhoria de Processos de Negócios**. São Paulo: Atlas, 2012.

OMG. **Documents Associated with Business Process Model and Notation™ (BPMN™) Version 2.0**, 2011. Available in: <http://www.omg.org/spec/BPMN/2.0/>. Accessed: 15 nov. 2016.

OTTENSOOSER, A. et al. Making Sense of Business Process Descriptions: An Experimental Comparison of Graphical and Textual Notations. **Journal of Systems and Software**, v. 85, n. 3, 2012.

PAIM, R. et al. **Gestão de processos:** pensar, agir e aprender. Porto Alegre: Bookman, 2009.

PICNIK. **The Picnik is over**, 2013. Available in: <http://www.picnik.com/>. Accessed: 15 nov. 2016.

PRILLA, M.; NOLTE, A. **Integrating Ordinary Users into Process Management:** Towards Implementing Bottom-Up, People-Centric BPM. CAiSE 2012. Berlin: Springer-Verlag. 2012.

RAO, L. TechCrunch. **Google Buys Flickr Photo Editor Picnik**, 2010. Available in em: <https://techcrunch.com/2010/03/01/google-buys-up-online-photo-editing-site-picnik/>. Accessed: 15 nov. 2016.

RECKER, J. Opportunities and constraints: the current struggle with BPMN. **Business Process Management Journal**, v. 16, n. 1, 2010.

ROLLAND, C.; NURCAN, S. **Business Process Lines to Deal with the Variability**. 47th Hawaii International Conference on System Sciences. Online: IEEE Xplore Digital Library. 2010.

ROSING, M. V. et al. Business Process Model and Notation - BPMN. In: ROSING, M. V.; SCHEER, A.-W.; VON SCHEEL, H. **The Complete Business Process Handbook:** Body of Knowledge From Process Modeling to BPM. Massachussets: Elsevier, v. 1, 2015. p. 429-453.

ROSING, M. V.; KIRCHMER, M. Focusing Business Processes on Superior Value Creation: Value-oriented Process Modeling. In: ROSING, M. V.; SCHEER, A.-W.; VON SCHEEL, H. **The Complete Business Process Handbook:** Body of Knowledge From Process Modeling to BPM. Massachussets: Elsevier, v. 1, 2015. p. 475-492.

RUSSEL, N.; VAN DER AALST, W. **Workflow Patterns:** The Definitive Guide. Massachussets: The MIT Press, 2016.

SANTOS, F. et al. **Atribuição de Responsabilidade aos Gestores em Processos de Negócio Transversais**. 7th Brazilian Symposium on Information Systems. Bahia: SBC. 2011.

SANTOS, F. et al. Aspect-oriented business process modeling: analyzing open issues. **Business Process Management Journal**, Bradford, v. 18, n. 6, 2012.

SCHERUHN, H.-J.; VON ROSING, M.; FALLON, R. Information Modeling and Process Modeling. In: ROSING, M. V.; SCHEER, A.-W.; VON SCHEEL, H. **In The Complete Business Process Handbook:** Body of Knowledge From Process Modeling to BPM. Massachussets: Elsevier, v. 1, 2015. p. 511-548.

SHARP, A.; MC DERMOTT, P. **Workflow Modeling:** Tools for Process Improvement and Application Development. 2. ed. Massachussets: Artech House, 2008.

SIAU, K.; TIAN, Y. **A semiotic analysis of unified modeling language graphical notations**. London: Springer-Verlag. 2008.

SWENSON, K.; VON ROSING, M. Phase 4: What is Business Process Management? In: ROSING, M. V.; SCHEER, A.-W.; VON SCHEEL, H. **The Complete Business Process Handbook:** Body of Knowledge From Process Modeling to BPM. Massachusetts: Elsevier, v. 1, 2015. p. 79-88.

TAVARES, F.; MARINHO, L. **AO-BPM 2.0: Modelagem de Processos Orientada a Aspectos. 2014.** Undergraduate Thesis, Federal University of the State of Rio de Janeiro. Rio de Janeiro. 2014.

TURETKEN, O. et al. The Effect of Modularity Representation and Presentation Medium on the Understandability of Business Process Models in BPMN. In: LA ROSA, M.; LOOS, P.; PASTOR, O. **Business Process Management**. Berlin: Springer-Verlag, v. 9850, 2016.

VAN DE BERG, K.; CONEJERO, J. M.; CHITCHYAN, R. **AOSD Ontology 1.0: Public Ontology of Aspect Orientation**. EU Network of Excellence on AOSD. Online. 2005.

VAN DER AALST, W. Business Process Management: A Comprehensive Survey. **ISRN Software Engineering**, Hindawi, v. 2013, n. 507984, 2013.

VAN DER AALST, W.; TER HOFSTEDE, A.; WESKE, M. Business Process Management: A Survey. In: VAN DER AALST, W.; WESKE, M. **Business Process Management**. Berlin: Springer-Verlag, v. 2678, 2003.

VAN DER AALST, W.; VAN HEE, K. **Workflow Management. Models, methods and systems**. Cambridge: The MIT press, 2000.

VANDERFEESTEN, I. et al. Quality metrics for business process models. In: FISCHER, L. **2007 BPM & Workflow Handbook:** Methods, Concepts, Case Studies and Standards in Business Process Management and Workflow. Florida: Future Strategies, v. 144, 2007.

WAHL, T.; SINDREAN, G. Analytical Evaluation of BPMN Using a Semiotic Quality Framework. In: SIAU, K. **Advanced Topics in Database Research**. Hershey: IGI Global, v. 5, 2006.

WEBER, B.; REICHERT, M. **Refactoring Process Models in Large Process Repositories**. Berlin: Springer-Verlag. 2008.

WESKE, M. **Business Process Management:** Concepts, Languages, Architectures. 2. ed. Berlin: Springer-Verlag, 2012.

WIKIPEDIA. **Top-down and bottom-up design**, 2016. Available in: <https://en.wikipedia.org/wiki/Top-down_and_bottom-up_design>. Accessed: 15 nov. 2016.

WIKIPEDIA. **Business Process Model and Notation**, 2016. Available in: <https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation>. Accessed: 15 nov. 2016.

WITTERBORG, H. et al. Weaving Aspects and Business Processes through Model Transformation. In: VILLARI, M.; ZIMMERMAN, W.; LAU, K.-K. **Service-Oriented and Cloud Computing**. Berlin: Springer-Verlag, v. 8745, 2014.