Pós-Graduação em Ciência da Computação

KAROLYNE MARIA ALVES DE OLIVEIRA

MABUP: Multi Level Autonomic BUsiness Process

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife
2015

Karolyne Maria Alves de Oliveira

MABUP: Multi Level Autonomic BUsiness Process

A PHD. THESIS PRESENTED TO THE CENTRO DE INFORMÁTICA OF UNIVERSIDADE FEDERAL DE PERNAMBUCO IN PARTIAL FULFILMENT OF THE REQUERIMENTS FOR THE DEGREE OF PHILOSOPHY DOCTOR IN CIENCIA DA COMPUTAÇÃO.

Area: Computer Science

Advisor: Prof. PhD. JAELSON FREIRE BRELAZ DE CASTRO

Recife
2015

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

O48m Oliveira, Karolyne Maria Alves de
MABUP: multi level autonomic business process / Karolyne Maria Alves de Oliveira. – 2015.
175 f.: il., fig., tab.

Orientador: Jaelson Freire Brelaz de Castro.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2015.
Inclui referências e apêndices.

1. Ciência da computação. 2. Gestão de processos de negócios. I. Castro, Jaelson Freire Brelaz de (orientador). II. Título.

004 CDD (23. ed.) UFPE- MEI 2019-015

Tese de Doutorado apresentada por **KAROLYNE MARIA ALVES DE OLIVEIRA** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**MABUP: Multi Level Autonomic Business Process**" orientada pelo **Prof. Jaelson Freire Brelaz de Castro**. APROVADA EM: 11/09/2015. Banca Examinadora formada pelos professores:

Prof. Ricardo Massa Ferreira Lima
Centro de Informática/UFPE

Prof. Robson do Nascimento Fidalgo
Centro de Informática / UFPE

Profa. Carla Taciana Lima Lourenço Silva Schuenemann
Centro de Informática / UFPE

Prof. Julio Cesar Sampaio do Prado Leite
Departamento de Informática / PUC/RJ

Prof. Luis Eduardo Galvão Martins
Departamento de Ciência e Tecnologia/ UNIFESP

Visto e permitida a impressão.
Recife, 11 de setembro de 2015.

Profa. Edna Natividade da Silva Barros
Coordenadora da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Acknowledgement

Thank, firstly, to God for every provision with which guided this work and all inspiration.

To my parents, Severino and Odete, for all the attention, affection and trust. I am sure their example of perseverance and especially love will guide me throughout life.

To my dear sister Ana Cristina, her husband Nielson and my niece Marianna, for being a reflection of kindness, humility and happiness.

To my family, to always understand my absences and be a mainstay at all hours.

To my religious community, Consolação Misericordiosa, my sisters in Cristo, Ana Veruska, sister Céu, and prayer group, which in times of trouble and difficulties always walked beside me.

Jaelson Castro, my advisor, to all collaboration and comprehension throughout the PhD.

The CIN lecturers, mainly to the lecturers Fernanda, Alexandre and Robson, for every contribution. The DSIC/UPV lecturers and professionals, especially Oscar Pastor, throughout trust and openness.

To my best friends forever, Ligia and Michelle, for companionship, love and strength.

To Guido Roiffe to be with me in this entire long trip and in ever difficult situation, for companionship and constant support.

To all the friends I made during the doctoral stage in Spain and in Brazil, especially Sergio, Murilo, Sebastian, Yeshica, for all the support, share emotions, difficult and happiness; To all LER group, specially Jessyka, Monique, Joao, Enyo and Bruno, throughout incentive and collaboration during the course of this work.

To all my co-workers of DATAPREV, especially Aluizio, Andre, Caroca, Ranieri, Rodrigo, Gustavo, Ana Esther, Paulo to the good laughs and encouragement.

Finally, everyone who contributed directly or indirectly to the success of this work.

Abstract

Business processes are becoming increasingly complex and heterogeneous in consequence of new stakeholders demands and technological advances. This calls for business processes that must be managed and executed in an autonomic manner in response to changes in its environmental context. Business Processes that are able to be self-managed are referred to as Autonomic Business Processes. A key challenge is to provide operational environment variability, understandability and scalability in modeling increasingly complex business processes. We achieve such goals by explicitly expressing environment variability through context-awareness, quality attributes which properly represent system parameters and the modularity through the use of multi-level approach for Autonomic Business Process. This novel approach, named Multi Level Autonomic Business Process - MABUP, provides four well-defined levels of abstraction to model business and operational knowledge namely, Organizational Level, Technological Level, Operational Level and Service Level; furthermore, an architecture to guide the autonomic management activity. Real examples were used to illustrate and evaluate our proposal through a simulation and controlled experiment. Results show that Multi Level Autonomic Business Process improves understandability and expressivity of autonomic features, modularization of Autonomic Business Process models and facility in business modeling. Furthermore, when used, can reduce time, cost and workload in its management.

***Keywords**: Autonomic computing. Business process management. Context-awareness. Multi-level modeling. Autonomic business process.*

Resumo

Os processos de negócios estão se tornando cada vez mais complexos e heterogêneos em consequência de novas demandas dos stakeholders e avanços tecnológicos. Isto exige que processos de negócios devam ser gerenciados e executados de forma autônoma em resposta a mudanças no seu contexto ambiental. Processos de Negócios que são capazes de autogestão são referidos como processo de negócio autonômico (do inglês Autonomic Business Process). Um dos principais desafios é fornecer variabilidade do ambiente operacional, compreensibilidade e escalabilidade na modelagem de processos de negócios cada vez mais complexos. Para alcançar tais objetivos, foi expresso explicitamente variabilidade do ambiente operacional, através da sensibilização do contexto; fez-se uso de atributos de qualidade, que representam adequadamente os parâmetros do sistema; e utilizou-se modularidade, através da utilização de abordagem multi nível para processo de negócio autonômico. Esta nova abordagem, denominada processo de negócio autonômico multi nível (MABUP – Multi Level Autonomic Business Process), oferece quatro níveis bem definidos de abstração para modelar negócios e conhecimento operacional, quais sejam: nível organizacional, nível tecnológico, nível operacional e nível de serviço. Além disso, uma arquitetura para orientar a atividade de gestão autonômica. Exemplos reais foram usados para ilustrar e avaliar nossa proposta através de uma simulação e experimento controlado. Os resultados mostram que processo de negócio autonômico multi nível melhora a entendimento e expressividade de características autonômicas, modularidade de modelos de processo de negócio autonômico e facilidade na modelagem de negócios. Adicionalmente, quando usado, reduz tempo, custo e carga de trabalho em sua gestão.

Palavras-chave: *Computação autonômica. Gestão de processos de negócio. Sensibilidade ao contexto. Modelagem multi nível. Processos de negócios autonômicos.*

List of Figures

List of Tables

List of Acronyms

ABP	*Autonomic Business Process*
BP	*Business Process*
BPD	*Business Process Diagram*
BPEL	*Business Process Execution Language*
BPM	*Business Process Management*
BPML	*Business Process Management Language*
BPMN	*Business Process Modeling Notation*
BPMS	*Business Process Management System*
CRM	*Customer Relationship Management*
NFR	*Non-Functional Requirements*
OMG	*Object Management Group*
QoS	*Quality of Service*
SLA	*Service Level Agreement*
SLO	*Service Level Objective*
SOA	*Service Oriented Architecture*
SoC	*Separation of Concerns*
WS	*Web Service*
XML	*Extensible Markup Language*

Contents

1 Introduction

This chapter presents an overview of the use of autonomic computing and business process management to help organizations to deal with complexity scenarios. Then, we present the problem that this thesis proposes to solve, as well as the objectives and relevance of the work; last but not least, we also describe the methodology used to achieve our goals.

1.1 CONTEXT

The enterprise environment is becoming quite competitive. For this reason, it is important to encourage solutions that can help organizations to survive increasingly dynamic scenarios. In this context, autonomic computing capabilities can be appropriate. The use of the word autonomic is derived from its biological origins. The control in the human body works in such a manner (self-regulating) that usually no humans' interference and consciousness are required. Likewise, the goal of autonomic computing is to create systems that run themselves, capable of high-level functioning while keeping the system's complexity invisible to the user (RAHMAN et al., 2011).

Autonomic Computing transfers maintenance responsibilities to the software itself. By automating tasks such as installation, healing or updating, system operation is simplified at the expense of increasing its internal complexity. A system with autonomic capabilities installs, configures, tunes and maintains its own components at run-time (CETINA; PELECHANO, 2013). Autonomic Computing Systems (ACS) are able to: (i) Self-configure: An autonomic application/system should be able configure and reconfigure itself under varying and unpredictable conditions; (ii) Self-heal: An autonomic application/system should be able to detect and recover from potential problems and continue to function smoothly; (iii) Self-optimize: An autonomic application/system should be able to detect suboptimal behaviors and optimize itself to improve its execution; and, (iv) Self-Protect: An autonomic application/system should be capable of detecting and protecting its resources from both internal and external attack and maintaining overall system security and integrity (PARASHAR; HARIRI, 2005).

The secondary properties of autonomic systems are: (i) Self-awareness: An autonomic system requires to know itself; (ii) Context-awareness: An autonomic system should be aware of its execution environment by exposing itself and discovering other systems in the environment; (iii) Openness: An autonomic system should be able to function in a heterogeneous environment and be implemented on open standards and protocols; (iv) Anticipatory: One critical property from the perspective of the users is that an autonomic system should be able to anticipate its needs and behaviors to act accordingly, while keeping its complexity hidden (RAHMAN et al., 2011).

Some works have proposed the use of Autonomic Computing (AC) concepts (HORN, 2001) to help to effectively manage enterprise systems and applications (T. Yu & Lin 2005). Indeed, a major application area for autonomic computing is aimed at freeing system administrators from the details of system operation and maintenance, improving robustness of systems and decreasing the total cost of ownership (GANEK; CORBI, 2003; KEPHART; CHESS, 2003).

On other hand, information and communications technology (ICT) solution should be aligned with business knowledge. The ICT function in every business is about capturing, processing, storing and distributing information or data. It is of paramount importance to align ICT with the business, which calls for an understanding of the business itself. Without such synergy, it is impossible to successfully achieve the desired outcome. A good starting point is to consider the multi-tier nature of an organization and how it fits into the rest of the business world. Typically the levels will follow a ladder pattern, representing the business since the corporate goals and processes to operational tasks (COLIN BEVERIDGE, 2006).

Business Processes and Business Process Management (BPM) are essential in many modern enterprises. They constitute organizational and operational knowledge and often perpetuate independently of personnel and infrastructure change (GREENWOOD; RIMASSA, 2007). Business process can be considered autonomic when tasks are automated and their maintenance can be predicted and executed with less human intervention (TERRES; NT; SOUZA, 2010). Autonomic Business Process (ABP) must have the capabilities to adjust to environment variation (context) occurred while a component node (services) is used (EVESTI; OVASKA, 2013).

In the Service-Oriented Computing (SOC) paradigm (PAPAZOGLOU et al., 2007), "services are autonomous and platform-independent computational elements that can be dynamically discovered, configured and (re)used" (TELES; LINS; ROSA, 2014). The combined use of BPM and SOC paradigms has the potential to empower organizations to dynamically manage their business processes, supported by flexible and reusable computational solutions that are capable to integrate complex and heterogeneous technologies.

In a service-based mission-critical system, adaptation is an activity with the objective of delivering an acceptable/guaranteed service based on Service Level Agreement (SLA) (YAU; YAN; HUANG, 2007). Basically, SLA is a service contract between a customer and a service provider that specifies the forwarding service a customer should receive. One of the key components in SLA is Service Level Objective (SLO), which specifies Quality of Service (QoS) metrics governing service delivery. Each SLA may include several SLOs, individually corresponding to a single QoS parameter related to quality factors. A SLO example is keeping the end-to-end response time in a specific range. Conditions in this model are post-conditions of changes. For example, how a substituted method, replaced using an aspect weaving/unweaving operation, will violate or satisfy a condition on the system throughput (SALEHIE et al., 2009).

In the domain of business applications, some solutions to incorporate autonomic capability have been proposed. For example, Autonomic Business Process (ABP) tries to cope with the changes in business environment by providing business processes with the autonomic capabilities (i.e., self-*) (TERRES; NT; SOUZA, 2010). However, it is worth noting that current approaches, are not concerned with the explicit representation in the business process model of the variability of the business environment and how the quality attributes can affect the adaptation. In these cases, autonomic managers contain the specification of adaptation and only consider operational deviations (CETINA; PELECHANO, 2013).

An explicit representation of business and operational environment variability through context-awareness and quality attributes is essential to provide more successful autonomic adaptation. Furthermore, it is important to overcome scalability problems and improving the understandability of ABP in complex scenarios.

1.2 PROBLEM

Business Process Management (BPM) lifecycle (VAN DER AALST; TER HOFSTEDE; WESKE, 2003) can be summarized in of four main stages, namely process design, system configuration, process enactment and diagnosis. It is a quite complex task to address autonomic capabilities from the first stage until the final realization (process enactment). A good starting point is to consider the multi-tier nature of an organization and how it fits into the rest of the business world. A business process is a net of activities that work together to achieve a defined goal, i.e., a defined business objective. In this sense, the representation of business process permeates different points of view since organizational level to operational level that represents the support of business needs. The depiction or grouping of the processes can be divided in three levels (i) strategic process- ensure that the underlying processes are meeting, and continue to meet, the specified organizational objectives; (ii) core processes – represents the core, or main, business activities of the organization; (iii) support processes - represents the non-core processes of the organization (JESTON; NELIS, 2014). In this sense, to help enterprises to face with processes that could support increasingly complex scenarios an evolution is incorporated with another way of management: the autonomic business process (ABP).

Strohmaaier and Yu in (STROHMAIER; YU, 2006) have presented the first attempt to apply autonomic computing principles to workflow management. They explained and showed that certain levels of autonomy can already be achieved with available techniques and introduced the concepts of the different degree of "autonomy" in workflow systems but did not present a new technique.

In order to deal with autonomic features, an interesting issue emerged when composing applications: the ability to bind and re-bind abstract activities to concrete services at run-time. Several researchers have addressed this issue. Lee et. al. (LEE et al., 2007a) discussed managing run-time adaptations in long running scientific workflows. In their work, adaptations have been described in terms of the functional decomposition of autonomic managers into monitoring, analysis, planning and execution components. Mosincat et al. (MOSINCAT; BINDER, 2008a) proposed fault tolerant execution of business processes executing dynamic binding of services, performing a process transformation before the execution, for using a selection component at run-time. However, these works did not represent the different facets of business needs. In this sense, the use of Separation of Concerns (SoC) and Modularization helps to guide appropriate adaptations. In Haupt et. al. (HAUPT; SUKHIJA; ZHUK, 2011), the workflow and the services comprising it are treated as managed resources controlled by hierarchically organized autonomic managers. Their work attempts to treat the complexity problem of autonomic workflows by using hierarchical workflow but, it does not explore any modularity technique neither how these different levels can guide different kinds of adaptations.

The analysis of the related work shows that there is a lot of interest in the development of systems and approaches able to deal, dynamically and automatically, with composition, binding, failures and other aspects of definition and execution of composed services that support business processes. Generally, the representation of (i) what business activities needs to be autonomic and monitored, (ii) to what autonomic feature and context and (iii) what to do in case of some quality deviation, i.e. how to operationalize non-functional requirements (NFR) (CARDOSO; ALMEIDA; GUIZZARDI, 2009), are not explored and aligned to business process specification. Furthermore, considering complex scenarios, these works do not explore any modularization technique to autonomic business process that helps to treat its management.

In this sense, **Table 1** presents a comparative evaluation of autonomic business process approaches showing that there is lack studies that combines features as self-awareness, context-awareness and modularization representation in business process models. Furthermore, some autonomic features could not be covered in some approaches.

Table 1 - Comparative of autonomic business process approaches

	Self-awareness representation in BP	**Context-awareness representation in BP**	**Modularization**	**Autonomic features represented in BP model**	**Autonomic features represented in autonomic manager**	**Autonomic features supported**
(LEE et al., 2007b)	X	-	-	-	X	Self-optimization and self-healing
(GREENWOOD; RIMASSA, 2007)	X	-	-	X	-	Self-configuration and self-optimization
(MOSINCAT; BINDER, 2008b)	X	-	-	-	X	Self-healing
(TRETOLA; ZIMEO, 2010)	X	-	X	-	X	Self-configuration
(HAUPT; SUKHIJA; ZHUK, 2011)	X	-	X	-	X	Self-healing

Emphasizing that some complex business processes are supported by services that are used by millions of users with simultaneous and absolutely stochastic access and the management of business processes is becoming increasingly complex, some challenges must be overcome, that is:

- *Can autonomic principles be applied to help organizations in dynamic business contexts?*
- *How to have expressive, yet understandable, autonomic business process models in increasingly complex scenarios?*

1.3 RELEVANCE

A multi-level approach to autonomic business process management using non-functional requirements (NFR) to represent service quality and contextualization to represent environment variation can highlight that:

- This approach brings benefits related to understandability of autonomic features in business process on the use of separation of concerns principles;
- Modularization can be applied to help the business analysts, software engineer and ICT (Information and Communication Technology) professionals to deal with expressivity of business and technological information in complex scenarios;
- Contextualization is important to be represented in autonomic environments to provide appropriate adaptation;

Other relevant aspects can be also mentioned:

- Encourage companies to express the operational information considering the different stages of business analysis;
- Foster the specification of autonomic features aligned to business processes in order to reduce the need for human intervention and complexity in its management;

1.4 OBJECTIVES

1.4.1 General Objective

The general objective of this work is to provide a multi level autonomic business process approach. This approach targets processes that need to be supported by service-based mission critical systems. In this sense, we want to explore concepts of modularity and separation of concerns in order to represent the different levels of business and operational information by doing so we hope to be able to separate the knowledge in the correct levels. We wish to rely on non-functional requirements to represent service quality, as well as to use contextualization to represent environment variation. We intend to use a MAPE (Monitor-Analyze-Plan-Execute) cycle that considers both the system (self) and the instrumented BPM (context).

1.4.2 Specific Objectives

- **Objective 1**: Investigate how the concepts used in autonomic computing could be applied in the management of business processes;

- **Objective 2**: Promote modularization and separation of concerns (SoC) in autonomic business process models;

- **Objective 3:** Define a meta-model that represents multi-level autonomic business process concepts;

- **Objective 4:** Architect the management of multi-level autonomic business process models;

1.5 CONTRIBUTIONS

Accordingly, we summarize the contributions of this work as follows:

- A systematic process that helps business analysts, software engineers and ICT (Information and Communication Technology) professionals to highlight the instrumentation of both business process model and system with autonomic capabilities;

- A meta-model with the concepts that must be considered in business process models to provide autonomic capabilities. This meta-model aims to improve the understandability of organizational, technological and operational knowledge using different levels of information to be analyzed by different kind of professionals;

- A systematic approach that exploits high variability in service-oriented system environment by using business process model to provide information about adaptations which rely on operationalization of non-functional requirements (NFR), i.e. define how to archive defined value for quality attributes;

- An architecture to support autonomic management of business process that considers both NFR and contexts to guide operational environment adjustments;

1.6 METHOD

The methodology process adopted in the development of this thesis can be decomposed in the follow steps:

1.6.1 Activity 1 – Bibliographic review

This research started with an exploratory study surveying and analyzing the business modeling bibliography to identify the state of the art of autonomic business process. Through this analysis, it was observed that the existing proposals for autonomic business process did not expressed any autonomic information in business process models and this information were specified in adaptation engines supporting services at run-time. Considering that autonomic features are not expressed in business process models, the following hypothesis was formulated: *The integration of autonomic features aligned to business process models would not benefit its management in complex scenarios.*

The remainder of this research was carried with the hypothetical deductive method. By analyzing the taxonomy of self-adaptation and business process management, we identified the gaps in autonomic business process approaches listed in the state of the art. In order to align adaptation information with business information, we identified the necessity of modeling techniques able to represent business, technology, operational and service information in process models. In this sense, we identified the need for a multi-level approach to represent autonomic business process, thus, there is evidence that the hypothesis may be rejected.

Assuming that such integration may indeed be beneficial, we defined a meta-model to represents the concepts that need to be incorporated to business process modeling languages. Furthermore, an architecture were outlined in order to consider the concepts listed.

1.6.2 Activity 2 – Data Collection and Sampling

Through case studies and feedback from analysts and students, we evaluated the proposed extensions to the business process modeling languages to express the variability of business processes operational environment in the context of autonomic systems. In order to achieve this goal, we collected quantitative and qualitative data on:

- a simulation that executed the process with real life information to analyze the benefits in execute multi-level autonomic business process;
- the expression power of autonomic features in business processes models;
- the use of separation of concerns and modularity technique to represent autonomic information in different levels;
- cognitive complexity and easiness of use through interviews / questionnaires about the complexity of the extended language to the context of autonomic systems.

1.6.3 Activity 3 – Data Analysis

The study analysis compares the collected data from simulation and a controlled experiment in order to assess if the hypotheses can be rejected. We use several statistical tests to refute null hypothesis.

1.7 Organization

Besides this introductory chapter, this thesis is structured as follows.

- **Chapter 2 – Theoretical Foundation**: Presents the background and related works.
- **Chapter 3 – MABUP – Multi Level Autonomic Business Process**: We introduce our multi-level autonomic business process management approach.
- **Chapter 4 – Experimental evaluation**: The evaluation of this approach through simulation and controlled experiment is described in this chapter.
- **Chapter 5 – Final Remarks –** This chapter presents the final remarks and the next steps of this research.

\

2 Theoretical Foundation

In this chapter, we introduce concepts about autonomic computing/self-adaptation as well as the use of business process modeling. Furthermore, we show some relevant works related to the use of this model to provide autonomic computing characteristics to software.

2.1 AUTONOMIC COMPUTING / SELF-ADAPTATION

The use of the word autonomic is derived from its biological origins. The control in the human body works in such a manner (self-regulating) that usually no humans' interference and consciousness are required. Likewise, the goal of autonomic computing is to create systems that run themselves, capable of high-level functioning while keeping the system's complexity invisible to the user (RAHMAN et al., 2011). Certain characteristics, known as self-CHOP properties are expected: (i) Self-Configuration; (ii) Self-Healing; (iii) Self-Optimizing; and, (iv) Self-Protecting (BABAOGLU, 2005; KEPHART; CHESS, 2003). **Figure 1** shows the hierarchy of the self-* properties (SALEHIE; TAHVILDARI, 2009).

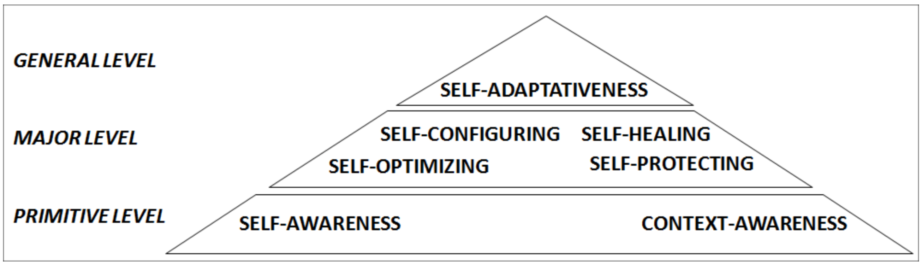


Figure 1 - Hierarchy of the Self-* Properties, adapted from (SALEHIE; TAHVILDARI, 2009)

Each level of this hierarchy is defined as follow:

- **General Level**: This level contains global properties of self-adaptive software. A subset of these properties, which falls under the umbrella of self-adaptiveness (OREIZY; MEDVIDOVIC; TAYLOR, 1998), consists of self-managing, self-governing, self-maintenance (KEPHART; CHESS, 2003), self-control (KOKAR; BACLAWSKI; ERACAR, 1999), and self-evaluating

[Laddaga 2006]. Another subset at this level is self-organizing (JELASITY; BABAOGLU, 2006), which emphasizes decentralization and emergent functionality(ies). A system with such a property typically consists of many interacting elements that are either absolutely unaware of or have only partial knowledge about the global system.

- **Major Level**: The IBM autonomic computing initiative defines a set of four properties at this level (HORN, 2001). This classification serves as the de facto standard in this domain. These properties have been defined in accordance to biological self-adaptation mechanisms (KEPHART; CHESS, 2003). For instance, the human body has similar properties in order to adapt itself to changes in its context (e.g., changing temperature in the environment) or self (an injury or failure in one of the internal organs).

- **Primitive Level:** Self-awareness, self-monitoring, self-situated, and context-awareness are the underlying primitive properties (HORN, 2001; SALEHIE; TAHVILDARI, 2005). Some other properties were also mentioned in this level, such as openness and anticipatory (PARASHAR; HARIRI, 2005), which are optional. The following list further elaborates on the details.
    - Self-Awareness (HINCHEY; STERRITT, 2006) means that the system is aware of its self-states and behaviors. This property is based on self-monitoring which reflects what is monitored.
    - Context-Awareness (PARASHAR; HARIRI, 2005) means that the system is aware of its context, which is its operational environment.

Self-adaptive software is expected to fulfill its requirements at run-time in response to changes. This objective can be achieved through monitoring the software system (self) and its environment (context) to detect changes, make appropriate decisions, and act accordingly.

In case, to appropriately elicit the essential requirements of self-adaptive software is very important to determine **when** and **where** a change is required, **how** it can be changed, and **what** the impacts of a change are on system attributes (LADDAGA, 2000). These dimensions (when, where, what, and how) are highlighted in the taxonomy of evolution (BUCKLEY; MENS; ZENGER, 2005). *When* corresponds to temporal properties; *Where* refers to the object of change and describe which artifact needs to be changed; *What* relates to system properties (qualities attributes); and *How* refers to change support (change models and the formality).

According to the taxonomy of self-adaptation (SALEHIE; TAHVILDARI, 2009), to provide adaptations is necessary to specify different facets as: (i) “object to adapt”; (ii) “realization issues”; (iii) “temporal characteristics” and (iv) “interaction concerns. These facets have a clear relationship with requirements questions (ie. when, where, what, and how). Although these facets cannot be mapped to the questions on a one-to-one basis, one or two questions in each facet emphasize different questions. **Figure 2** illustrates the hierarchy of this taxonomy.

Object to Adapt deals with *where* and *what* aspects of the change once in this facet is defined which node, component or application must be monitored and what operations should be done to realize the correct adjust. In fact, both sets of *what* and *where* questions are covered in the developing and operating phases. The definition of this facet is subdivided in:

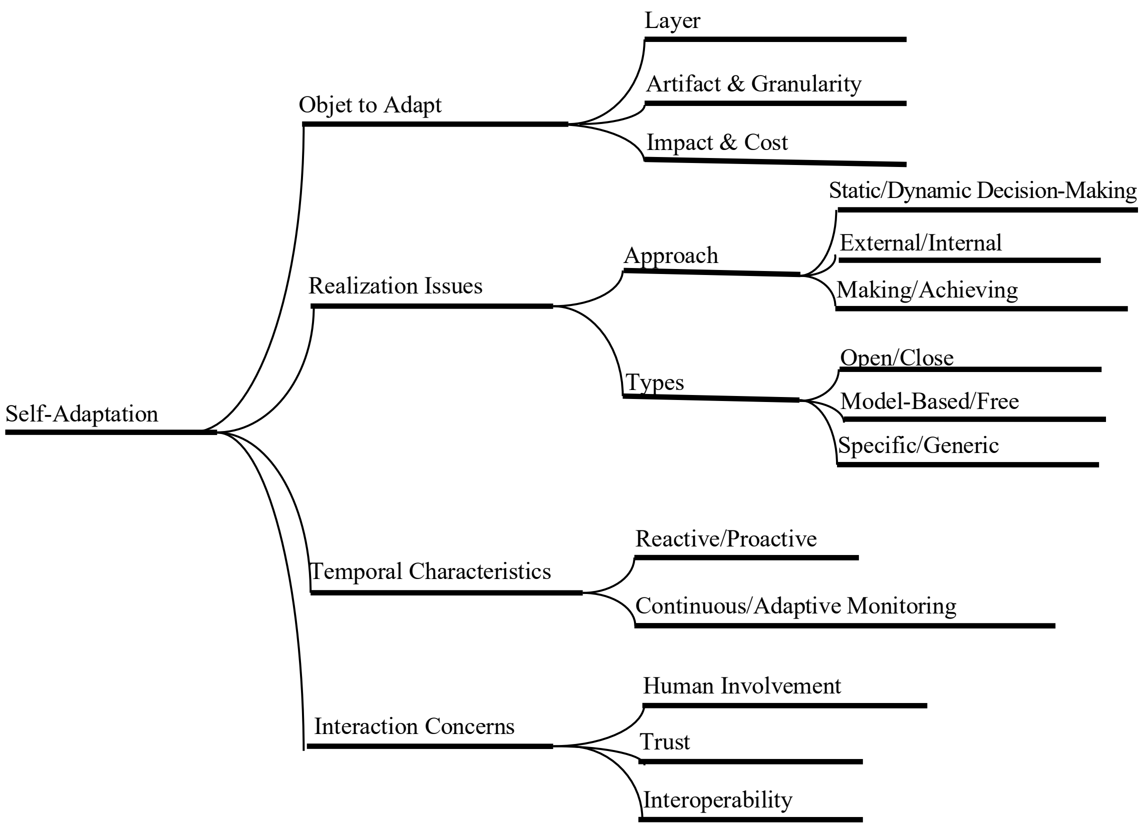


Figure 2 - Taxonomy of Self-Adaptation, adapted from (SALEHIE; TAHVILDARI, 2009)

- Layer: Determines which layer of the software system (i.e., where) must be assured based on the requirements; which layer of the system can be changed and needs to be changed; and what adaptation actions can be applied to different layers. For example, in application-layer adaptation, changes usually have direct impact on the user, and consequently, they may require the user's explicit approval and trust, in middleware adaptation changes can represents architectural adaptations.

- Artifact and Granularity: Determines which artifact(s) and at which level of granularity can/needs to be changed; what artifact, attribute, or resource can/needs to be changed for this purpose; Depending of the artifact or granularity, adaptation can change the modules or the architecture and the way they are composed. An application can be decomposed into services, methods, objects, components, aspects, and subsystems depending on the architecture and technologies used in its implementation. Each of these entities, as well as their attributes and compositions, can be subject to change, and therefore, adaptation can be applied in fine or coarse levels of granularity.

- Impact & Cost: The impact describes the scope of after effects, while cost refers to the execution time, required resources, and complexity of adaptation actions. This facet is related to what the adaptation action will be applied to, and partly to how it will be applied. An example is the case of having a load-balancing action that routes requests through duplicate components or servers. Another noteworthy point is that strong actions are mostly composite, and may contain several weak/strong actions. For example, changing the architecture may require redeployment of some components and changing a few parameters.

Realization Issues deals with *how* the adaptation can/needs to be applied. This issue is categorized into approach and type classes.

- Adaptation Approach. Indicates how to incorporate adaptivity into the system. The following sub-facets can be identified:
    - Static/Dynamic Decision-Making: This sub-facet deals with how the deciding process can be constructed and modified. In the static

option, the deciding process is hard-coded and its modification requires recompiling and redeploying the system or some of its components. In dynamic decision-making, policies, rules or Quality of Service (QoS) definitions are externally defined and managed, so that they can be changed during run-time to create a new behavior for both functional and non-functional software requirements.

  - External/Internal Adaptation: From a different perspective, the adaptation can be divided into two categories with respect to the separation of the adaptation mechanism and application logic. Internal approaches intertwine application and the adaptation logic. External approaches use an external adaptation engine (or manager) containing adaptation processes.
  - Making/Achieving Adaptation: self-adaptivity can be introduced into software systems using two strategies. The first strategy, Making, is to engineer self-adaptivity into the system at the developing phase. The second strategy, Achieving, is to achieve self-adaptivity through adaptive learning which implies in having artificial intelligence and adaptive learning.

- Adaptation Type. It specifies whether the adaptation is open or closed to new alternatives, whether it is domain specific or generic, and whether it is model-based or model-free.
  - Close/Open Adaptation: A close-adaptive system has only a fixed number of adaptive actions, and no new behaviors and alternatives can be introduced during run-time. On the other hand, in open adaptation, self-adaptive software can be extended, and consequently, new alternatives can be added, and even new adaptable entities can be introduced to the adaptation mechanism.
  - Model-Based/Free Adaptation: In model-free adaptation, the mechanism does not have a predefined model for the environment and the system itself. On the other hand, in model-based adaptation, the mechanism utilizes a model of the system and its context.

- Specific/Generic Adaptation: Some of the existing solutions address only specific domains/applications, such as a database (e.g., IBM SMART project [IBM SMART]). However, generic solutions are also available, which can be configured by setting policies, alternatives, and adaptation processes for different domains (e.g., Accord). This type addresses *where* and *what* concerns in addition to *how*, because the specific type only focuses on an adaptation of artifacts or attributes of a particular part of the software system.

Temporal Characteristics deals with issues regarding *when* artifacts can/need to be changed. The following sub-facets can be identified:

- Reactive/Proactive Adaptation: In the reactive mode, the system responds when a change has already happened, while in the proactive mode, the system predicts when the change is going to occur. This issue impacts the detecting and the deciding processes.

- Continuous/Adaptive Monitoring: Captures whether the monitoring process (and consequently sensing) is continually collecting and processing data vs. being adaptive in the sense that it monitors a few selected features, and in the case of finding an anomaly, aims at collecting more data. This decision affects the cost of the monitoring and detection time.

Interaction Concerns consists of interacting with humans and/or other elements/systems. The facet is related to all four of the *where-when-what-how* questions as well as the "*who*" question. The following sub-facets can be identified:

- Human Involvement: This facet is related to the question of "who" the agent of change. In self-adaptive software, human involvement can be discussed from two perspectives. First, the extent to which the mechanism is automated, and second, how well it interacts with its users and administrators. The second view addresses the quality of human interaction to either express their expectations and policies, or to observe what is happening in the system. According to this view, human involvement is essential and quite valuable for improving the manageability and trustworthiness of self-adaptive software.

- Trust: It is a relationship of reliance, based on past experience or transparency of behavior. One view of trust is security and how much human or other systems can rely on self-adaptive software systems to accomplish their tasks. This view is linked to assurance and dependability.

- Interoperability Support: Self-adaptive software often consists of elements, modules, and subsystems. Interoperability is always a concern in distributed complex systems for maintaining data and behavior integrity across all constituent elements and subsystems.

We argue that to guide business-oriented adaptations, we must model it in a well-defined level of granularity once it determines the impact of the adaptation. Considering organizational knowledge, in our approach the impact affects services-oriented systems with coarse grained operational adaptation.

We assume the static decision-making diagnosis with an external autonomic engine that continuous monitors the system environment and act according a model-based mechanism. The main idea is to reduce human involvement in business process management.

2.1.1 Feedback Control

We advocate the use of feedback control (or closed loop control) (SALEHIE et al., 2009), as the mechanism to support autonomic characteristics into a business process. We argue that self-adaptive software is a closed-loop system with feedback from the self and the context. The Autonomic Computer System's building block, named autonomic manager, constantly interacts with managed element in order to maintain its equilibrium in face of incoming perturbations. The MAPE cycle (Monitor-Analyze-Plan-Execute), represented in **Figure 3,** is an implementation of the classical feedback control technique (KEPHART; CHESS, 2003).

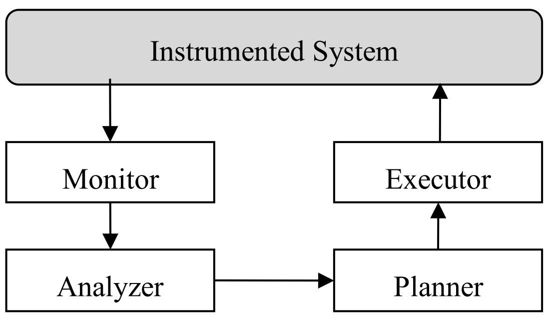


Figure 3 - Closed Loop Control Mechanism

The process of designing a control system generally involves many steps (DOYLE; FRANCIS; TANNENBAUM, 1990). A typical scenario can be resumed as follows:

1) Model the characteristic of system to be controlled;
2) Analyze the model and determine its properties;
3) Decide on performance specifications;
4) Design a controller to meet the specifications, if possible; if not, modify the specifications or generalize the type of controller sought;
5) Simulate the resulting controlled system, either on a computer or in a pilot plant;
6) Choose hardware and software and implement the controller.

When a closed loop control is in action, the system is managed while interacts with external environment. When environment provokes distortions in system behavior, the monitor (also named "sensor") will detect them, through constant comparing carried over a set of metrics output by the system. If distortion between compared parameters exceeds pre-defined thresholds, analyzer alerts planner component, passing information extracted from the metrics read. The planner will plan an intervention (adjustments) upon the system, in order to set it back to accepted regime. The plan is put in practice through actuators that will interact with the system, changing some of its parameters of execution.

In this thesis, we will explore feedback control mechanism to provide information that can guide the execution of the multi-level autonomic business process.

2.1.2 Autonomic Computing Features

Kephart and Chess (2003) have presented the notion of autonomic computing principles, ie. self-configuration, self-optimization, self-healing and self-protection (KEPHART; CHESS, 2003), as follow:

- **self-configuration:** autonomic systems configure themselves automatically in accordance with high-level policies—representing business-level objectives, for example—that specify what is desired, not how it is to be accomplished. For example, when a context happens into an autonomic accounting system, it will automatically identify and take into account the composition and configuration of its environment. It will register itself and its capabilities so that other components can either use it or modify their own behavior appropriately.

- **self-optimization**: complex middleware or database systems may have hundreds of tunable parameters that must be set correctly for the system to perform optimally, yet few people know how to tune them. Such systems are often integrated with other, equally complex systems. Consequently, performance-tuning one large subsystem can have unanticipated effects on the entire system. Autonomic systems will monitor, experiment with, and tune their own parameters and will learn to make appropriate choices about keeping functions or outsourcing them. They will proactively seek to upgrade their function by finding, verifying, and applying the latest updates.

- **self-healing**: autonomic computing systems will detect, diagnose, and repair localized problems resulting from bugs or failures in software and hardware, probably through a regression tester. Using knowledge about the system configuration, a problem-diagnosis would analyze information from log files, possibly supplemented with data from additional monitors that it has requested. The system would then match the diagnosis against known software patches (or alert a human programmer if there are none), install the appropriate patch, and retest.

- **self-protection:** Despite the existence of firewalls and intrusion- detection tools, humans must at present decide how to protect systems from malicious attacks and inadvertent cascading failures. Autonomic systems will be self-protecting in two senses. They will defend the system as a whole against large-scale, correlated problems arising from malicious attacks or cascading failures that remain uncorrected by self-healing measures. They also will anticipate problems based on early reports from sensors and take steps to avoid or mitigate them.

According to Kephart and Walsh (2004) (KEPHART; WALSH, 2004), the manner to represent high-level policies to adapt systems is the use of **utility functions** that map any possible state of a system to a scalar value, as shown in **Figure 4**. They can be obtained from Service Level Agreement (SLA), preference elicitation or simple templates. They are very useful representation for high level objectives once the value can be transformed and propagated among agents to guide system behavior.

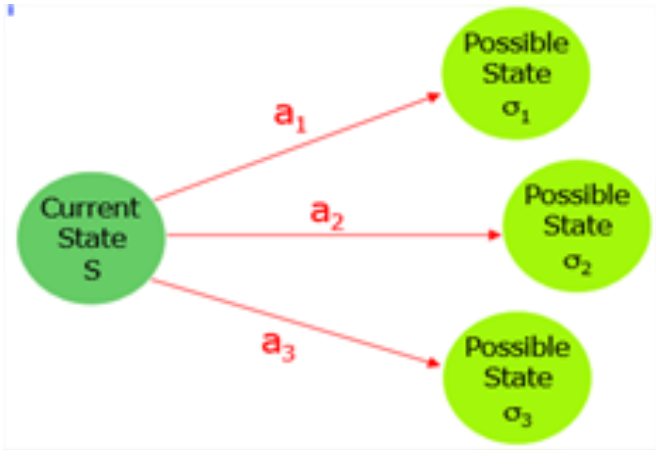


Figure 4 - Possible states of a system (KEPHART; THOMAS, 2011)

Utility function, in the context of autonomic computing, is a way to capture the preference of an agent i. A utility can be interpreted as: U_i (s), where i is the agent, s is a state of the world and assign a value, for example 7. So, the utility of the agent i in the state s is 7. The maximum value for a utility is the best choice. In this sense, policies can be managed in four steps according the relationship of the terms involved in the specification:

- **Elicit** a utility function U(**S**) and express it in terms of services attributes **S;**
- **Model** how each attribute $\mathbf{S_i}$ depends on controls **C** and observables **O,** as **S(C;O),** where *controls* refers to values that represents a system configuration and *observables* are quality values. How *observables* changes, set to C values to maximize U'(C;O). Values can be obtained from experiments, learning or theory;
- **Transform** from service utility U to resource utility U' by substitution:

$$U(S) = U(S(C;O)) = U'(C;O) \qquad (2.1)$$

- **Optimize** resource utility. An observable O changes, set **C** to values that maximize U'(**C;O**).

$$C^*(O) = \text{argmax}_C U'(C;O) \qquad (2.2)$$

$$U'^*(O) = U'(C^*(O);O) \qquad (2.3)$$

For example, we could specify that the response time (S_1) of the web server should be under 2s (O_1), while that of the application server under 1s (O_2). The autonomic manager uses internal rules (i.e., knowledge (C) which each combination of states is quantified to add or remove resources as necessary to achieve the desirable state. The utility function could take as input the response time for web and application servers and return the utility of each combination of web and application server response times. The maximum value for each state is the best choice.

Although the use of utility enjoys of a well-grounded theory, the challenge is that this approach is very hard to be elicited from humans in complex scenario once every aspect that influences the decision must be quantified and not aligned to business information. Moreover, it considerers only services quality attributes and is restricted to autonomic engines.

We argue that autonomic features should be aligned to business knowledge through contextualization. In this sense, we claim that concepts used in autonomic systems must be represented in Business Processes Models.

2.2 BUSINESS PROCESS MANAGEMENT

In the 1980s there was a considerable focus on Total Quality Management (TQM). This was followed in the early 1990s by Business Process Reengineering (BPR) as promoted by Hammer and Champy (1990). BPR had a chequered history, with some excellent successes as well as failures.

Following BPR in the mid- and late 1990s, Enterprise Resource Planning (ERP) systems gained organizational focus and became the next big thing. These were supposed to deliver improved ways for organizations to operate, and were sold by many vendors as the ‘solution to all your problems’. The ERP systems certainly did not solve an organization’s process issues, nor make the processes as efficient and effective as they could have been. Towards the end of the 1990s and in the early 2000s, many Customer Relationship Management (CRM) systems were rolled out with extensive focus on the customer view and customer experience. While this provided focus on the front office, it did not improve the back-office processes. More recently, Six Sigma has started to come into its own.

In 2000, the definition of BPM involved:

1. <u>Reach the strategic goals</u> of an organization by <u>improving</u>, <u>managing</u> and <u>controlling</u> their essential business processes (GESTON; NELIS, 2014);

2. Disciplined approach to identify, design, execute, document, measure, monitor, control and improve business processes, automatized or not, to reach the intended results aligned and consistent with strategic goals of an organization (APBMP CBOK, 2012);

3. Set of techniques, technologies, and governance mechanism that offer a standard and scalable way to manage the processes of an organization.

According to Rademakers & Barrez (2012), creating a fully functional business process involves five steps, often referred to collectively as the BPM life cycle, shown in **Figure 5**.

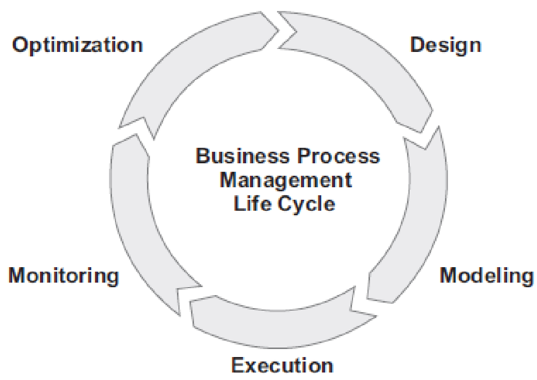


Figure 5 - The five steps of business process management discipline: design, modeling, execution, monitoring and optimization (RADEMAKERS; BARREZ, 2012)

Each of these five steps represents an important development phase in implementing a successful process solution:

- Design—The first step consists of activities that define the business process: identifying high-level activities, discussing possible organizational changes, defining service level agreements, and specifying process details such as actors, notifications, and escalations.

- Modeling—In this step, the business process is fully specified and validated. The process flow is formalized, for example, by using BPMN; additionally, process variables are defined and candidate services that can be used to execute an activity are identified. To validate the business process, "what-if" scenarios are performed with process simulation.

- Execution—The modeled business process is implemented in a business process application, often using a business process management system (BPMS) such as Activiti, jBPM, Bonita. You still need to add technical details to the business process before you can execute it. The process is implemented with a process language like WSBPEL or BPMN 2.0.

- Monitoring—The processes are monitored for business goals that are defined by key performance indicators (KPIs). Examples of KPIs are "the average number of orders received in a day should be at least 30" and "the time to send a proposal to a customer based on a web inquiry shouldn't exceed eight hours."

- Optimization—Based on new insights, changing business requirements, and monitoring results, the implemented business processes will need to be optimized. When the optimization phase is done, the business process goes into the design phase again and the cycle is completed.

The BPM life cycle shows that implementing business processes is an ongoing process due to the ever changing business environment and need for optimization. How long it takes to walk through all five steps of the BPM life cycle depends greatly on the business environment and the ability of a business to execute. In some businesses, it may take years to complete the cycle; in others, it can be done in weeks or even days.

2.2.1 BPM Architecture

Business process management architecture is related to three perspectives: Techniques, Technologies and Governance Mechanisms. Techniques involves how to model, analyze, design a business process, measure its performance and realize transformations according the results obtained. Technologies are related to the technological support to the modeling and execution of business process. Finally, governance mechanisms involve the maintenance of the business process during the time; in this sense, can be realized in different ways, eg. process management lifecycle, process office, business process maturity model (BPMM) and business process management guidelines. **Figure 6** presents an overview of the concepts related to business process management when involve these three perspectives.

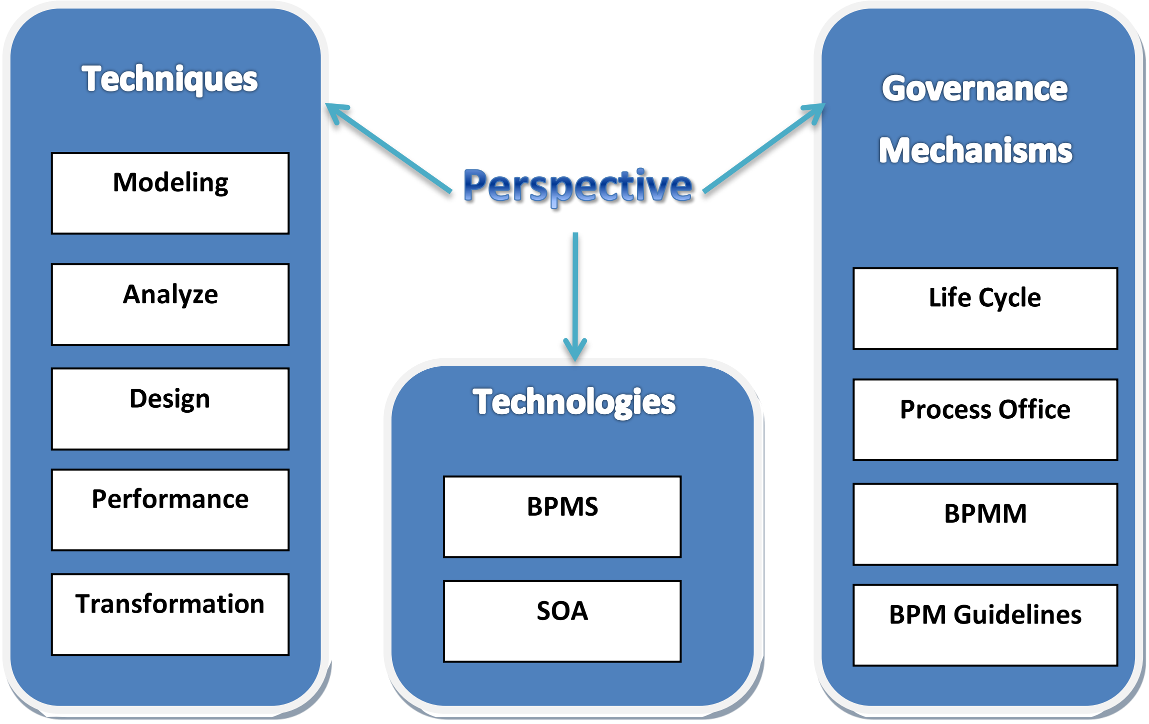


Figure 6 - Business Process Management

According to BPM architecture, this thesis explores techniques to guide autonomic business process modeling, analysis, design and performance. This approach also uses technologies as Business Process Management System (BPMS) and Service-Oriented Architecture (SOA). As governance mechanism, we explored business process life cycle. Next sections, we will detail existing business process modeling approaches that can be further extended to incorporate autonomic capabilities. We evaluate existing BPMS that can be used to automate business process models and facilitate governance of multi-level autonomic business process life cycle.

2.3 Business Process Modeling

Business Processes and Business Process Management (BPM) are essential in many modern enterprises. They constitute **organizational** and **operational** knowledge and often perpetuate independently of personnel and infrastructure change (GREENWOOD; RIMASSA, 2007). We claim that autonomic computing principles can be adapted to help organizations survive in dynamic business scenarios.

Business process model is becoming quite popular among business analysts. The notation is based on the representation of the working process of an organization in terms of activities performed and flow of actions and process data. The activities and tasks of a process can be performed by people or automatically by software. This section presents some of the most important languages for business process that can be explored to be extended with autonomic capabilities.

2.3.1 Unified Modeling Language

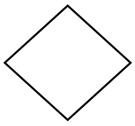
The Unified Modeling Language (UML) is a diagramming language or notation to specify, visualize and document models of object-oriented software. The UML is standardize by the Object Management Group (OMG) and is the industry standard for graphically describing software (OMG, 2010a). The UML has eleven different types of diagrams which, although heterogeneous, are described in a uniform framework via a meta-model that formally describes the abstract syntax of all types of diagrams and a notation guide that defines the concrete syntax for the elements the meta-model.

UML model was initially developed to model software systems, however, certain types of diagram as the diagram of activities, can also be used for modeling business processes. In the activity diagram, a business process can be described by an activity that consists of sequencing of nodes, based on control-flow and object-flow.

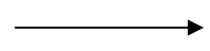
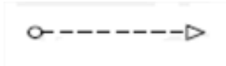
A prominent feature of UML is the possibility to extend their vocabulary through definition of profiles, which are a series of stereotypes. Stereotypes define new elements from existing ones (eg. an activity), by the addition of new properties that are chosen for a specific context.

2.3.2 Business Process Modeling Notation (BPMN)

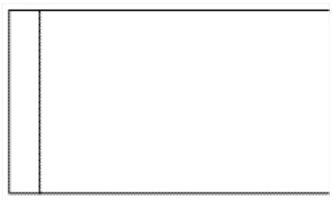
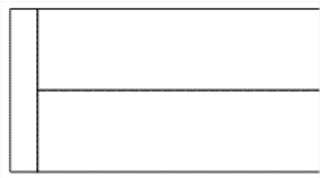
BPMN stemmed from the effort of modeling tool suppliers to standardize process modeling notations (WHITE; MIERS, 2008). The first version of BPMN was based on workflow notation that is derived from several others as UML activity diagram. BPMN is based on the activities flow and artifacts in the organizational process. **Figure 7** shows core set of BPMN elements to represent a business process.

Object	Figure
Events	
Activities	
Gateway	

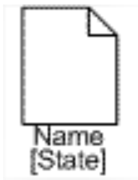
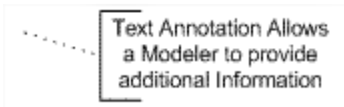
(a) Flow objects

Object	Figure
Sequence Flow	
Message Flow	
Association	

(b) Connecting objects

Object	Figure
Pool	
Lane	

(c) Swinlanes

Object	Figure
Data Object	
Group	
Annotation	

(d) Artifacts

Figure 7 - Core Set of BPMN Elements (ABPMP, 2009)

BPMN describes a set of language constructs. Basically it comprises as follow:

a) Flow objects- are the main graphical elements that define the behavior of business processes. These are divided in three types of objects: events, activities and gateway.

- **Events**: An event is something that "happens" during the course of a business process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result). There are three types of events, based on when they affect the flow: Start, Intermediate and End.

- **Activities**: An activity is a generic term for work that a company performs. An activity can be atomic or non-atomic. The types of activities that are a part of a process model are: Process, Sub-Process, and Task.

- **Gateways**: A gateway is used to control the divergence and convergence of a sequence flow. Thus, determine branching, bifurcations, merge and joining of paths. Internal markers indicate the type of behavior.

b) Connection Objects - There are three ways to connect the flow objects, that are: Sequence Flow, Message Flow and Association.

- **Sequence Flow:** The flow sequence is used to indicate the order in which the activities are being carried out in a process.

- **Message Flow**: A Message Flow is used to show the flow of messages between two participants who are prepared to receive and send messages.

- **Association:** An association is used to associate information with Flow Objects. The arrow indicates the direction of the association flow.

c) Swinlane - Swimlanes are entities that realize the grouping of elements. They are used to organize a BPMN diagram according to the responsible for executing activities. There are two ways of grouping the main modeling elements through swimlanes: pools and lanes.

- **Pool:** A pool represents a participant in a process that also can acts as a swim lane or as a container for partitioning a set of pools of other activities, usually in the context of B2B situations.

- **Lane:** A lane is a sub-division inside a pool and goes over the entire length of the pool, both vertically and horizontally. The lanes are used to organize and categorize activities.

d) Artifacts - Artifacts represent the inputs and outputs of the activities a process. These are composed of three categories of representative elements. They are:

- **Object Data:** Elements required or produced by activities connected to them through associations.
- **Group:** It is an element whose purpose is the documentation or analysis.
- **Annotations:** Element used to add additional information to the readers of the diagram.

The current version of BPMN (OMG, 2010b) includes four diagrams: Process Diagram, Collaboration Diagram, Conversation Diagram and Choreography Diagram. They describe the process in different views: the process itself (process diagram) and high level communication between participants (conversation diagram).

2.3.3 Web Service Business Process Execution Language (WS-BPEL)

The WS-BPEL language (Web Services Business Process Execution Language) is a standard proposed by OASIS (Organization for the Advancement of Structured Information Standards) for the execution of business processes, describing how the relationship between various Web Services participants of a composition occur (JORDAN; ALVES, 2007). It allows the composition of Web Services in order to perform more complex functionalities.

BPEL essentially extends imperative programming language (eg C) with builders to implement Web Services. Programming with WS-BPEL is similar to the programming of existing programming languages, as it offers certain types of constructors such as repeating structures, conditional structures, variables and functions. This fact allows the business process be seen as an algorithm.

A composition of services specified in WS-BPEL is seen as a service, that is, those that execute a business process defined in WS-BPEL are not aware of the various services required to perform a certain operation. Being a web service, the composition needs to have its interface based on standard WSDL (CHRISTENSEN et al., 2001), message exchange with other web services via SOAP (BOX et al., 2000) standard and can be registered in a UDDI repository for consultation (CURBERA et al., 2002). The WS-BPEL assumes that the services participating in the composition are described using WSDL.

2.3.4 Communication Analysis

Communication Analysis proposes undertaking information system analysis from a communicational perspective (ESPANA et al., 2009). It offers a requirements structure and several modeling techniques for BPM and requirements specification. Among these techniques, the Communicative Event Diagram and the Event Specification Templates are primary for conceptual model derivation. The Communicative Event Diagram is intended to describe business processes from an organizational perspective. A communicative event is a set of actions related to information (acquisition, storage, processing, retrieval, and/or distribution), that are carried out in a complete and uninterrupted way, on the occasion of an external stimulus. For each event, the actors involved are identified, as well as the corresponding communicative interactions and the precedence relations among events.

Moreover, Communication Analysis offers guidelines to allow the identification of communicative event as well as modularization of business process. The Unit Criteria is used to guide the information systems modeling action (GONZALEZ; ESPANA; PASTOR, 2009). It consists of three rules:

- *Trigger unity*, that presumes that the event occurs as a response to an external interaction by an actor that triggers organizational reaction;
- *Communication unity*, that describes that each and every event involves providing new meaningful information;
- *Reaction unity*, that outlines that an event triggers an Information System (IS) reaction, which is a composition of synchronous activities. Events are asynchronous among each other. This criterion defines a temporal encapsulation.

2.4 BUSINESS PROCESS MANAGEMENT SYSTEMS (BPMS)

According Chang (2005), Business Process Management System (BPMS) is a new class of software that allows organizations to devise process-centric information technology solutions. Process-centric means BPMS solutions are able to integrate people, systems, and data. Organizations that utilize BPMS to accomplish IT–enabled business process change will gain the following capabilities:

1. Closer business involvement in designing IT–enabled business process solutions;
2. Ability to integrate people and systems that participate in business processes;
3. Ability to simulate business processes to design the most optimal processes for implementation;
4. Ability to monitor, control, and improve business processes in real time;
5. Ability to effect change on existing business processes in real time without an elaborate process conversion effort.

There are some open source process engines, eg. Activiti, jBPM, Bonita, Intalio; so, it is important to analyze details about the open source frameworks. BAINA and BAINA (2013) and RADEMAKERS and BARREZ (2012) present a comparative study on these workflow systems.

- Activiti: under an Apache Licence, Activiti is a lightweight workflow and Business Process Management (BPM) Platform targeted at business people, developers and system admins. Its core is a fast and solid BPMN 2 process engine for Java. It is a full open-source solution and distributed under the Apache license. Activiti runs in any Java application, on a server, on a cluster or in the cloud. It integrates with Spring, it is extremely lightweight and based on simple concepts. Activiti acknowledges that executable business processes must be applicable as a component in everyday software development. That's why Activiti spends a lot of effort in making sure that it can be used very easily in *every* Java environment. This includes the cloud as soon many applications will be written for the cloud. Activiti's primary purpose and focus is to implement the general-purpose process language BPMN 2.0 but also to support any custom process languages built on top of it.

- *JBoss BPM or jBPM*—An open source process engine that first supported the custom jPDL process language, but, because version 5.0 supports BPMN 2.0, the jBPM project has merged with the JBoss Drools project (an open source business-rule management framework) and replaced Drools Flow as the rule flow language for the Drools framework. Under Apache/Eclipse/MIT Licence, jBPM is the first historical reference in the domain of open source BPM and workflow tools, it has been promoted byJBossas the leader of open source business process engines. jBPM is based on the BPMN 2.0 specification and supports the entire life cycle of the business process (from authoring through execution to monitoring and management). jBPM offers open-source business process execution and management, including (i) embeddable, lightweight Java process engine, supporting native BPMN 2.0 execution (ii) BPMN 2.0 process modeling in Eclipse (developers) and the web (business users) (iii) process collaboration, monitoring and management through the Guvnor repository and the web console (iv) human interaction using an independent WS-HT task service; (v) tight, powerful integration with business rules and event processing.

- *BonitaSoft*—An open source process engine that provides support for the BPMN 2.0 process language. The main differentiators of BonitaSoft are the large set of supported elements and the integrated development environment. Under a GPL for BonitaOpen Solution (BOS) studio & a LGPL for BOS execution engine, Bonita was initially developed by Bull, and it has been recently outsourced under the control of an independent company BonitaSoft. Bonita is available under two packages an open source community edition and a commercial one. Bonita's purpose is to offer a simple, intuitive and graphical solution that support development of ready to use BPMN process based applications.

Activiti and jBPM have a lot in common: they are both developer-oriented process engine frameworks built around the concept of a state machine. Because jBPM 5 also implements the BPMN 2.0 specification, a lot of similar functionality can be found. There are also a number of differences between Activiti and BonitaSoft as presented in **Table 2**.

Table 2 - Difference between Activiti, jBPM and Bonita

Description	Activiti	jBPM	Bonita
Community members	Activiti has a base team consisting of Alfresco employees. In addition, companies like Spring Source, Fuse Source, and Mule-Soft provide resources on specific components. There are also individual open source developers committing to the Activiti project.	jBPM has a base team of JBoss employees. In addition, there are individual committers.	Bonita is very popular and well adopted by the open source developers.
Spring support	Activiti has native Spring support, which makes it easy to use Spring beans in your processes and to use Spring for JPA and transaction management.	jBPM has no native Spring support, but you can use Spring with additional development effort.	Bonita has no native Spring support.
Business rules support	Activiti provides a basic integration with the Drools rule engine to support the BPMN 2.0 business rule task.	jBPM and Drools are integrated on a project level, so there's native integration with Drools on various levels.	Bonita has integration with Drools rule engine to support the BPMN 2.0 business rule task.
Additional tools	Activiti provides modeler (Oryx) and designer (Eclipse) tools to model new process definitions. The main differentiator is the Activiti Explorer, which provides an easy-to-use web interface to start new processes, work with tasks and forms, and manage running processes. In addition, it provides ad hoc task support and collaboration functionality.	jBPM also provides a modeler based on the Oryx project and an Eclipse designer. With a web application, you can start new process instances and work with tasks. The form support is limited.	BonitaSoft provides a large set of connectivity options to a wide range of third party products. This means it is easy to configure a task in the developer tool to connect to SAP or query a particular database table. Also provides a tool-based solution where you can click and drag your process definition and forms.
Project	Activiti has a strong developer and user community with a solid release schedule of two months. Its main components are the Engine, Designer, Explorer, and REST application.	jBPM has a strong developer and user community. The release schedule isn't crystal clear, and some releases have been postponed a couple of times. The Designer application is (at the moment of writing) still based on Drools Flow, and the promised new Eclipse plugin keeps getting postponed.	Bonita has an open source community edition and a commercial one.

Since it is of paramount importance for companies to have their processes automated in order to be more efficient, the use of BPMS is increasingly relevant. Furthermore, the adoption of autonomic computing principles aligned with business process management is an opportunity to deal with dynamic business contexts and reduce human intervention. For this reason, in this work we investigated the use of BPMS to deal with autonomic features. We elected Activiti once it is easy to use, has a good documentation and an active community to consult.

2.5 Autonomic Computing Principles and Business Process with Non-Functional Requirements and Context

Business process modeling involves the accurate capture of operational behavior and the associated process constraints. Even with the technological support to business processes, there is a belief that self-* properties are related to software quality factors. For example, Salehie and Tahvildari discuss the potential links between these properties and quality factors (SALEHIE; TAHVILDARI, 2005).

Non-functional requirements (NFR) consists in constraints, softgoals and quality attributes that must be assure in the system. This may include a variety of properties including performance expectations, policy constraints, and security controls. These characteristics are generally identified at some point after the business process modeling exercise (PAVLOVSKI; ZOU, 2008) .

The term quality requirements and software system attributes are also used to represent NFRs. NFRs are considered as the constraints or qualifications of the operations and place restrictions on the product being developed, the development process or specify external constraints that the product must exhibit.

There are 252 types of NFRs in the literature, **Table 3** presents the detail definition and attributes of the top five used NFRs (MAIRIZA; ZOWGHI; NURMULIANI, 2010).

Table 3 - The Most Commonly Considered NFRs

NFR	Definition	Attribute
Performance	requirements that specify the capability of software product to provide appropriate performance relative to the amount of resources needed to perform full functionality under stated conditions	response time, space, capacity, latency, throughput, computation, execution speed, transit delay, workload, resource utilization, memory usage, accuracy, efficiency compliance, modes, delay, miss rates, data loss, concurrent transaction processing
Reliability	requirements that specify the capability of software product to operates without failure and maintains a specified level of performance when used under specified normal conditions during a given time period	completeness, accuracy, consistency, availability, integrity, correctness, maturity, fault tolerance, recoverability, reliability, compliance, failure rate/critical failure
Usability	requirements that specify the end-user-interactions with the system and the effort required to learn, operate, prepare input, and interpret the output of the system	learnability, understandability, operability, attractiveness, usability compliance, ease of use, human engineering, user friendliness, memorability, efficiency, user productivity, usefulness, likeability, user reaction time
Security	requirements that concern about preventing unauthorized access to the system, programs, and data; or any illicit activity	modifiability, analyzability, changeability, stability, maintainability compliance
Maintainability	requirements that describe the capability of the software product to be modified that may include correcting a defect or make an improvement or change in the software	confidentiality, integrity, availability, access control, authentication testability, understandability

We argue that in logical level of business process model the adaptations must result in quality attributes defined according NFR Catalog (MAIRIZA; ZOWGHI; NURMULIANI, 2010). In order to demonstrate such relationships, it is better to analyze how a well-known set of quality factors defined in the ISO 9126-1 quality model (ISO/IEC 9126-1, 2001) are linked to major and primitive self-* properties. Considering aspects discussed in (GANEK; CORBI, 2003) and (TORRES; MARTINS, 2014), **Table 4** presents a consolidation of NFR and Autonomic Computing Principles.

Table 4 - NFR and Autonomic Computing Principles

NFR	Autonomic Computing Principles			
	Self-Configuring	Self-Healing	Self-Optimize	Self-Protect
Maintainability	X	X		
Functionality	X		X	X
Portability	X			
Usability	X			
Reliability	X	X		X
Efficiency			X	

Recently, works on process mining showed how management of processes, in various domains and engineering of supporting systems, could be guided by models extracted from the event logs that are recorded during process operation. For example, the value of queue mining to a specific operational problem (i.e. some quality to be assured) of online delay prediction is presented in (SENDEROVICH et al., 2014). In (MAGGI et al., 2014), authors provided a framework that takes into account both the sequencing of activities as well as data associated to the execution of each activity. A validation of the framework using a real-life log demonstrates that recommendations generated based on the framework have a promising level of accuracy when sufficient support is available.

The benefits related to the use of NFR joined with BPM have been acknowledged in the last years. NFRs have been applied to help in the design of business process models through extensions of business process modeling languages, allowing for richer analysis and operationalizations of the process model (PAVLOVSKI; ZOU, 2008) (CARDOSO; ALMEIDA; GUIZZARDI, 2009). However, many works do not consider the variability in their solutions (SANTOS et al., 2011). On the other hand, the context of a business process is the set of environmental properties that affect business process execution. Therefore, these properties should be taken into account when designing a business process. If context is analyzed when modeling a business process, then identification of all its variants (relevant states of the world in which the business process is executed) and definition of how the business process should be executed in them are facilitated (LUIS et al., 2010).

In order to express autonomic principles in business processes we argue the incorporation in their models of the concepts of NFR and contexts. Next section we will describe modularization techniques in order to proper represent different concerns in business process models.

2.6 MULTI LEVEL BUSINESS PROCESS AND SEPARATION OF CONCERNS

Business Process and Business Process Management (BPM) are essential to many modern enterprises as they constitute **organizational** and **operational** knowledge which often is perpetuated independently of personnel and infrastructure change (GREENWOOD; RIMASSA, 2007). The evolution of business process modeling has been strongly influenced by the relationship with new technologies such as Service-Oriented Architecture (SOA) that enables new application scenarios in which small loosely pieces of functionality are published, consumed and combined with other functions over a network. A key feature of SOA is a two-level model to implement global enterprise systems where business functions are built as combinations of services. Despite the close relationship between the business process view in technology and organization, this concept is considered differently and for different goals by software engineers and business analysts (ABETI et al., 2008).

The emergence of new enterprises requires open and interoperable technologies supporting their processes. **Figure 8** identifies the some points of view representing the three main perspectives for business modeling:

- **Enterprise view**: At this level, the models deal with the organization, strategies, business rules, business domain, etc. This perspective is commonly used by business analysts, which focus on business improvement;
- **System view**: the models concern the organization, the internal business processes, the business entities, the systems, the architectures, etc. It is the

perspective of Requirements Engineers, which concentrate on understanding the need of the stakeholders ;

- **Execution view**: At this level, the goal is to define an executable model for business processes. It is the software engineers' perspective for system design and implementation.

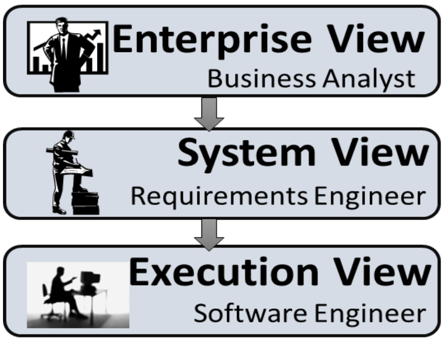


Figure 8 - The three views for business process modeling, adapted from (ABETI et al., 2008)**.**

In the case of an autonomic system, it is expected the support of configuration, healing, optimization and protection (CHOP) properties, for example being responsible for automating maintenance tasks such as installation, healing or updating. System operation is simplified at the expense of increasing internal complexity. For this reason, in order to treat autonomic features it is crucial to consider operational knowledge and align with business processes using a well-defined level of granularity. In doing so, it would be possible to determine the impact of the adaptation as well as to strive for understandability and scalability in complex business scenarios. We our approach we will rely with these different perspectives.

Furthermore, it is important to represent how to model different points of view of business process and how to separate each level considering both service **quality factors** and environment **context** information.

Modularization helps to treat the complexity of large-size models (REIJERS, 2008). Once the management of business process models can be realized at different levels of granularity, there is a tradeoff between monitoring granularity and diagnostic precision. The finest level of monitoring granularity is at the functional level where all leaf level tasks are monitored. The disadvantage of fine-grained level monitoring is the cost of the high monitoring and diagnostic overhead. Coarser levels of granularity only monitor higher-level business goals. In these cases, less complete and high-level monitoring data is generated, leading to multiple competing diagnoses. Both monitoring and diagnostic overheads are lower. The disadvantage is that if requirements denials are found, multiple diagnoses are returned, each pinpointing possible failures (WANG; MYLOPOULOS, 2009).

For this reason, the use of modularization techniques is imperative. Additionally, to provide autonomic information in business process models, it is important to represent technological and operational points of view. In order to manage the business process according its service of support we argue that we must specify the services level.

On the other hand, there are many reasons for the adaptation in context-aware workflow. Adaptations can be customization, correction, and optimization. Customization refers to the adaptation that is triggered by the change of context related to user's need such as the need for new services or the replacement of an original service. The workflow is adapted to accommodate the new requirement or context. During execution, a service that is bound to a task might become unavailable and the workflow needs to adapt to compensate the failure. This corrective type of adaptation deals with faults and usually involves rolling back the failed task and restarting a part of workflow. The adaptation can also be employed to optimize the performance such as to improve workflow completion time or to detect and solve bottlenecks in the workflow (SMANCHAT; LING; INDRAWAN, 2008).

The **separation of concerns** principle is a means to manage multifaceted domains. It is a generic problem solving heuristic that addresses the constraints of a problem domain individually and then attempts to combine the partial solutions as a single and nearly optimal overall solution to the problem (CAETANO; SILVA; TRIBOLET, 2012). Furthermore, the problem solving activity itself needs to yield partial solutions that are compostable. As such, these abstraction strategies attempt to reduce complexity by factoring out details so that one can focus on a few concepts at a time (WEBER, 1978). Abstraction paradigms used in systems modeling and design include classification, generalization and decomposition (OMG, 2010b). In this sense, it is important to represent different autonomic information in different perspectives. In this work, we use SoC to separate different categories of autonomic information to be included in business process models.

To evaluate the use of these concepts in business process models to provide autonomic principles, it is necessary to perform an experimentation to analyze if the proposal will archive the planned goals. Next section we will describe the experimental evaluation process that we have used in this thesis.

2.7 EXPERIMENTAL EVALUATION

Experiment is created to test a theory or **hypothesis**. In the design of the experiment, a number of **treatments** (values that the studied variable can take) over which researchers have control are observed. The experiment is performed to test the relationship between a treatment and the outcome to draw conclusions about the cause-effect for which we stated a hypothesis. There are two kinds of variables in an experiment, **independent** and the **dependent variables**. Dependent variables (or response variables) revel the effect of the changes in the independent variable (factor), which can be manipulated and controlled (eg. study the effect of a new design method on the productivity of personnel. In this case, productivity is de dependent variable. The design method, experience of the personal, tool support and environment are independent variables). A treatment is one particular value of a factor. The choice of treatment, and which levels the other independent variable shall have, is part of experiment design (see **Figure 9**).

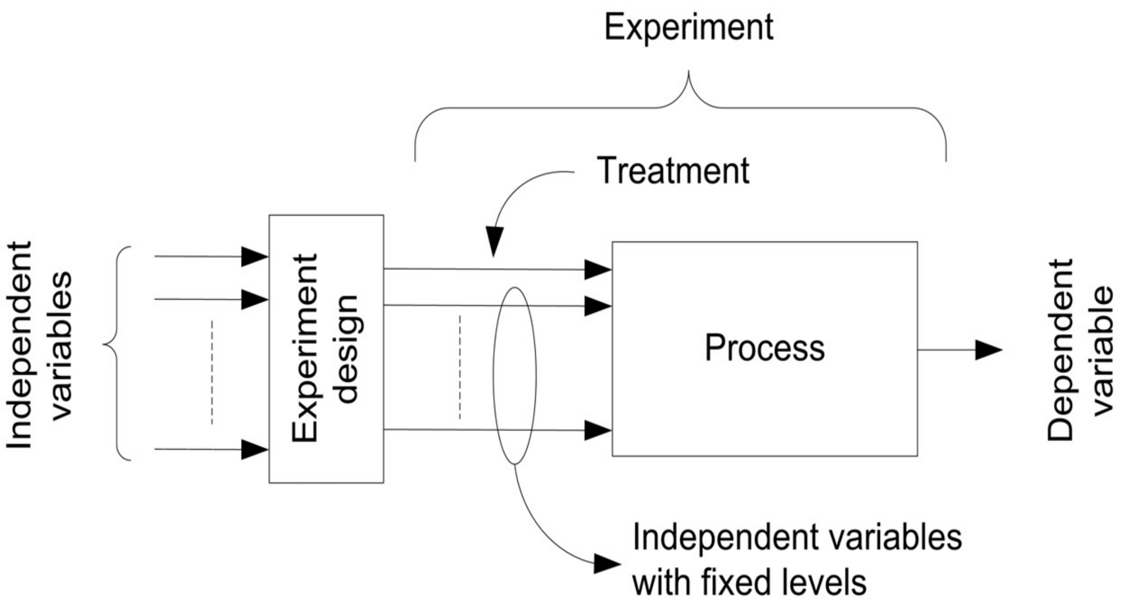


Figure 9 - Illustration of an experiment (WOHLIN et al., 2000).

The treatments are being applied to the combination of **objects** and **subjects**. An object can, for example, be a model that shall be reviewed with different inspection techniques. In its turn, the people that apply the treatment are called subjects.

An experiment consists of a set of **tests** where each test is a combination of treatment, subject and object (eg. A *test* can be that person N (*subject*) uses the new development method (*treatment*) for developing program A (*object*)). The number of tests affects the experimental error and provides the opportunity to estimate the mean effect of any experimental factor.

2.7.1 Experiment Design

The first activity to perform an experiment is the definition of what must be done. In this sense, the hypothesis has to be stated clearly. Furthermore, the objective and goals must be defined. The goal is formulated from the problem to be solved. In order to capture the definition, some aspect should be considered: object of the study (what is studied), purpose (what is the intention), quality focus (which effect is studied), perspective (whose view) and context (where the study is conducted).

In the next step, it is important to determine both independent (input) and the dependent (output) variables, as well as the values the variables actually can take. This also includes determining the measurement scale, which puts constraints on the method that later the researchers can apply for statistical analysis. The subjects of the study are identified.

Furthermore, experiment is designed which includes choosing a suitable experiment design including, for example, randomization of subjects and to prepare the instrumentation of the experiment. In this moment, researchers must identify and prepare suitable objects, develop guidelines if necessary and define measurements procedures. Finally, it is important to consider the question validity of the results that will be expected. Validity can be divided into four major classes: (i) Internal Validity: concerned with the validity within the given environment and the reliability of the results; (ii) External Validity: evaluates how general the finding are, ie. if the results from an experiment are valid outside the actual context in which it was run; (iii) Construct Validity: evaluates if the treatment reflects the cause construct and the outcome provides a true picture of the effect construct; and (iv) Conclusion Validity: concerned with the relationship between the treatment and the outcome.

2.7.2 Experiment Strategy

After design the experiment, it is necessary to define how to operationalize it. The strategy consists in principle of three steps: (i) Preparation: concerned with preparing subjects as well as the material needed (eg. Data collection forms). The participants must be informed about the intention of the experiment, consent and commit with it; (ii) Execution: ensure that the experiment is conducted according to its plan and design which includes data collection; (iii) Validation: ensure that the collected data is correct and provide a valid picture of the experiment.

2.7.3 Analysis and Statistical Tests

The data collected during the operation provide the input of this activity. The first step in the analysis is to understand the data using descriptive statistics to provide a visualization of the data. Descriptive statistics helps to understand and interpret the data informally.

The first analysis helps to consider whether the data set should be reduced, either by removing data points or by reducing the number of variables by studying if sine if the variables provide the same information.

After having removed data points or reduced data set, researchers are able to perform hypothesis test, where the actual test is chosen based on measurement scales, values on the input data and the type of results that the researchers are looking for. The objective of hypothesis testing is to see if it is possible to reject a certain null hypothesis H_0, based in a sample from some statistical distribution. That is, the null hypothesis describes some properties of the distribution from which the same is drawn and the experimenter wants to reject that these properties are true with a given significance.

Tests can be classified into parametric and non-parametric. Parametric tests are based on a model that involves a specific distribution. In most cases, it is assumed that some of the parameters, involved in a parametric test, are normally distributed and can be measured at least on an interval scale. Some tests can be used to verify if the distribution is normal or not, for example, Anderson-Darling test. If a parametric cannot be measured in at least an interval scale this means generally cannot be used. In this case, there are a wide range of non-parametric tests available. The choice between parametric and non-parametric statistical methods considers the applicability and power of the tests. The power of parametric tests is generally higher than for non-parametric tests. The main tests are described below:

- **T-test:** One of the most often used parametric tests. Used to compare two sample means. That is, the design is one factor with two treatments;
- **U-test or Mann-Whitney:** Non-parametric alternative to the t-test;
- **F-test:** This is a parametric test that can be used to compare two sample distribution;
- **Paired t-test:** A t-test for a paired comparison design;
- **UtWilcoxon:** Non-parametric alternative to paired t-test;
- **Sign test:** Non-parametric alternative to paired t-test. The sign test is a simpler alternative to the Wilcoxon test;
- **ANOVA (Analysis Of VAriance):** A family of parametric test that can be used for designs with more than two levels of a factor;
- **Kruskal-Walls:** Non-parametric alternative to ANOVA in the case of one factor with more than two treatments;
- **Chi-2 (Chi-square):** This is a family of non-parametric test that can be used when data are in form of frequencies.

Table 5 presents the difference of these tests.

Table 5 - Overview of parametric / non-parametric tests for different design

Design	Parametric	Non-parametric
One factor, one treatment		Chi-square Binominal test
One factor, two treatments, completely randomized design	F-test T-test	U-test (Mann-Whitney or Wilcoxon rank sum test) Chi-square
One factor, two treatments, paired comparison	Pared t-test	U-test (Mann-Whitney or Wilcoxon rank sum test) Sign test
One factor, more than treatments	ANOVA	Kruskal-Walls Chi-square
More than two factor	ANOVA	

An important aspect of this activity is the interpretation of the data to determine from the analysis whether the hypothesis was accepted or rejected. This forms the basis for decision-making and conclusions concerning how to use results from the experiment, which includes motivation for further studies.

The last step to be done is concerned with presenting and packing of the findings by documentation of the results, discussion and threats to validity.

2.8 CHAPTER SUMMARY

In this chapter, we outlined concepts related to autonomic computing, business process modeling and how autonomic computing principles can be related to business process models. We argue that in order to provide adaptability, it is important to elicit the dimensions when, where, what, and how that are highlighted in the taxonomy of evolution. Likewise, in the taxonomy of self-adaptation these dimensions are represented in four different facets: "object to adapt"; "realization issues"; "temporal characteristics" and "interaction concerns".

Some of the most well-known business processes modeling languages were introduced. As mentioned before, it is very important to model the organizational knowledge in a well-defined level of modularity once it determines the impact of the adaptation while also provide understandability and scalability in complex business scenarios. For this reason, the use of modularization techniques is imperative. Furthermore, to provide autonomic information in business process models, it is important to represent technological and operational points of view. To manage the business process according its service of support we argue that we must specify the services level. All this vision show BPM expressed under different perspectives to improve understandability of autonomic information. The objective of this thesis is present an approach able to deal with concepts that are necessary to provide autonomic features in business process management. In order to verify this hypothesis, it is important to perform an experimental evaluation as descripted above. Next chapter we will present our proposal to autonomic business process.

3 MABUP – Multi Level Autonomic Business Process

In this section, we present our Multi Level Autonomic Business Process - MABUP approach, which consists of four well-defined abstraction levels. It relies on a closed-loop mechanism to provide system adaptation. Our framework considers both instrumented business process model (with context-awareness information) and Services (with self-awareness).

3.1 Instantiating the Taxonomy of Self-Adaptation to Multi Level Autonomic Business Process

As discussed in section 2.1, in order to proper specify autonomic features, it is necessary to define the dimensions **when** and **where** a change is required, **how** it can be changed, and **what** the impacts of a change are on system attributes. Furthermore, it is important to characterize "object to adapt", "realization issues", "temporal characteristics" and "interaction concerns". Considering these facets, we analyzed the taxonomy of self-adaptation to guide multi level autonomic business processes as detailed:

- **Object to be adapted:** this facet deals with *where* and *what* aspects. In this sense, we have to represent both: a well-defined level where an automated business process task can be monitored according a SLA; and operational level that represents what to do if occurs a deviation in the non-functional requirements attributes. The definition of this facet is divided in:
    - **Layer:** In business process models scenarios it is important to define levels that represents *where* and *what*, ie. a process that provide a well-defined level of business process to be monitored and operational level that represents what to do if some deviation occurs;
    - **Artifact and Granularity:** We define that the artifact and granularity that must be adapted is an automated atomic task supported by services with a SLO.
    - **Impact and cost:** The scope and cost of adaption is described in terms of operational tasks that must be supported by services able to modify resources.
- **Realization Issues:** this facet deals with *how* the adaptation is to be applied. To provide autonomic business process, we argue that it is important to provide a

level that present tasks to execute actions with the objective to return the system to an optimum state. The adaptation approach used in this work is dynamic decision-making, since we rely on autonomic business process policies definitions when are externally defined by services available to support changes in operational environment. Our approach has a making adaptation once self-adaptivity is planned by engineer and introduced into the services at the developing phase. Furthermore, we did not predict any adaptive learn in the scope of this work. We advocate the use of close adaptation that has only a fixed number of adaptive actions, and no new behaviors and alternatives can be introduced during run-time.

- **Temporal Characteristics** deals with issues regarding *when* artifacts can/need to be changed. Our MABUP approach can have reactive adaptation according with quality attribute threshold and context modeled that are continuous monitored.
- **Interaction Concerns** consists of interacting with humans and/or other elements/systems. The facet is related to all four of the *where-when-what-how* questions as well as the "*who*" question. MABUP approach aims to have no human involvement in adaptation and high interoperability.

3.2 Systematic Process

We have separated all these concerns in four abstraction levels of granularity and extended MAPE engine to consider the information of the instrumented business process model.

In **Figure 10**, we depict an overview of MABU approach that includes two main stages: (i) the **modeling phase** exploits the modeling of four abstraction levels: Organizational Level, Technological Level, Operational Level and Service Level. We used four abstraction levels to provide separation of concerns in autonomic business process more specifically express organizational knowledge (organizational level), the definition of autonomic business processes including monitored tasks (technological level) and operational tasks that should be executed if some deviations occurs (operational level), lastly, the service knowledge (service level); (ii) the **management phase** includes a MAPE module that uses the modeled autonomic business process and the metrics provided by the systems to guide the adaptation.

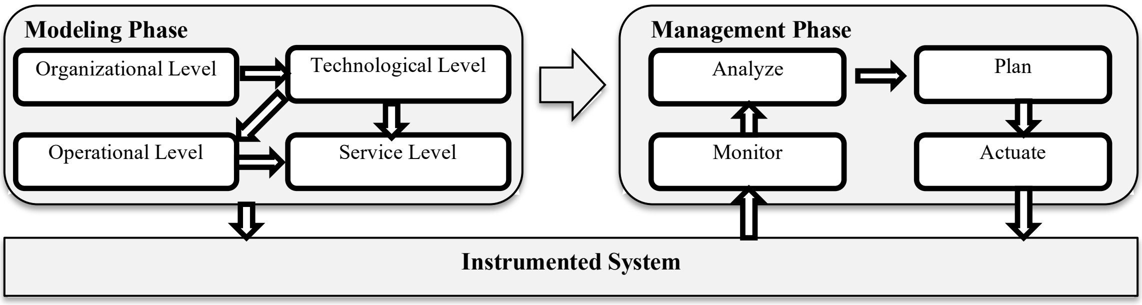


Figure 10 - Overview of the modeling and management phases of MABUP

In order to help the incorporation of autonomic taxonomy definitions in the management of business process models, we now present a process that emphasize the information and actions necessary to instrument both business process model and its correspondent system. The process consists of seven steps (see **Figure 11**):

- **Step S1 – Define Organizational Level of Business Process Model:** This step incorporates the use of techniques to provide a well-defined level of business process model, which representation is immune to technological changes. The objective is to provide only the organizational behavior in a defined level of modularity and identify mission critical activities, i.e. represent activities able to process an entry and return an output without express how technology can impact in this processing. As result, we obtain the initial business process model. By using BPMN, it is necessary to modularize in sub-process the processing of the same information;

- **Step S2 – Define Autonomic Processes:** Considering the initial model and the SLA, this step indicates that the SLA should be guaranteed to some activities of the initial model and after refinement in tasks identify the points that should be treated in an autonomic manner, ie. define tasks that should be monitored and adapted if some deviation occurs. This treatment involves also operational tasks that should be executed in that case. As result, we obtain the autonomic business process. Considering the existence of monitored tasks and operational tasks that are supported by services, this approach is adequate to be aligned with service-oriented architectures. In MABUP approach, we defined different levels to represent autonomic processes: Technological level, which represents a refinement of mission critical activities and indicate what tasks should be monitored; and Operational Level, which indicate the operational tasks;

- **Step S3 – Link Services with Autonomic Processes**: Once the autonomic tasks are defined, this step establishes the link between these tasks and their corresponding services. In this case, there are two kind of services: the one that support business task and the other to support adaptation actions. It is necessary to build services that can transform operational environment. In MABUP approach, the service information is represented in the Service Level;

- **Step S4 – Define metrics**: Considering SLA and NFR Catalogue, this step identifies the information (ie. context and NFR) that can affect the process and the parameters to trig the operational intervention to manage its environment. As result, we obtain a scored autonomic business process, ie. a model with the representation of metrics that must be collect to guide adaptations. In MABUP approach, this information is represented in Operational Level;

- **Step S5 – Define Services QoS**: Considering the metrics specified in the model, this step establishes the instrumentation of services with this information to provide a log with the NFR information. As result, we obtain the services with QoS attributes. In MABUP approach, this information is represented in Service Level;

- **Step S6 – Monitor System at Run-Time**: Considering the scored autonomic business process and the QoS defined, this step defines the continuous monitoring of the system. The idea is to rely on metrics previously identified to check for anomalies. In MABUP approach, this step is realized in the management phase;

- **Step S7 – Choose Autonomic Processes**: According to what is monitored, it is necessary to analyze deviations. If it occurs the system should plan and choose the operational tasks that should be executed to return the system to its optimum state. In MABUP approach, this step is realized in the management phase;

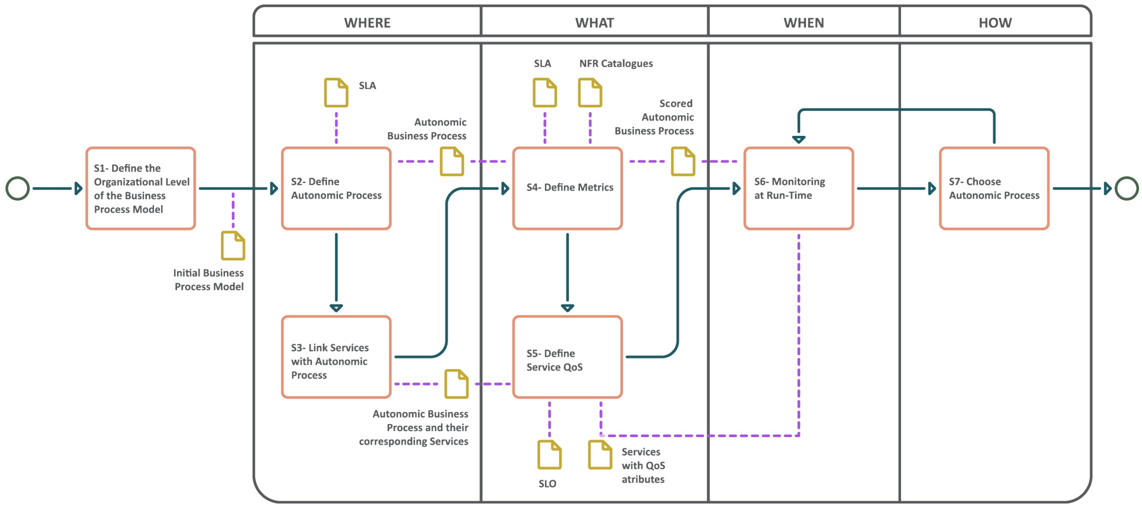


Figure 11 - Steps to define multi-level autonomic business process and the instrumentation of the system

In the sequel we present our Multi-Level Autonomic Business Process approach, named MABUP. As mentioned before, our framework takes in consideration both context information and quality attributes of the system. A running example will be used to illustrate our approach and we will use BPMN language to model the knowledge.

3.3 Running Example

The example consists of a large system that requires to be managed in an autonomic manner. Specifically we treated the characteristic of Self-Optimization. We examine the CAGED (General Register of Employed and Unemployed), a project under the Ministry of Labor and Employment of Brazil (MTE) and governed by law 4923/65. It supports the submission of monthly declarations of change of company's employees due to dismissal or admittance (CAGED movements). The deadline for submission is the 7th day of every month. The data submitted are related to the previous month (i.e. its competence). The declarations are processed to generate operational and statistical data for the ministry of labor and employment, as can be illustrated in **Figure 12**.

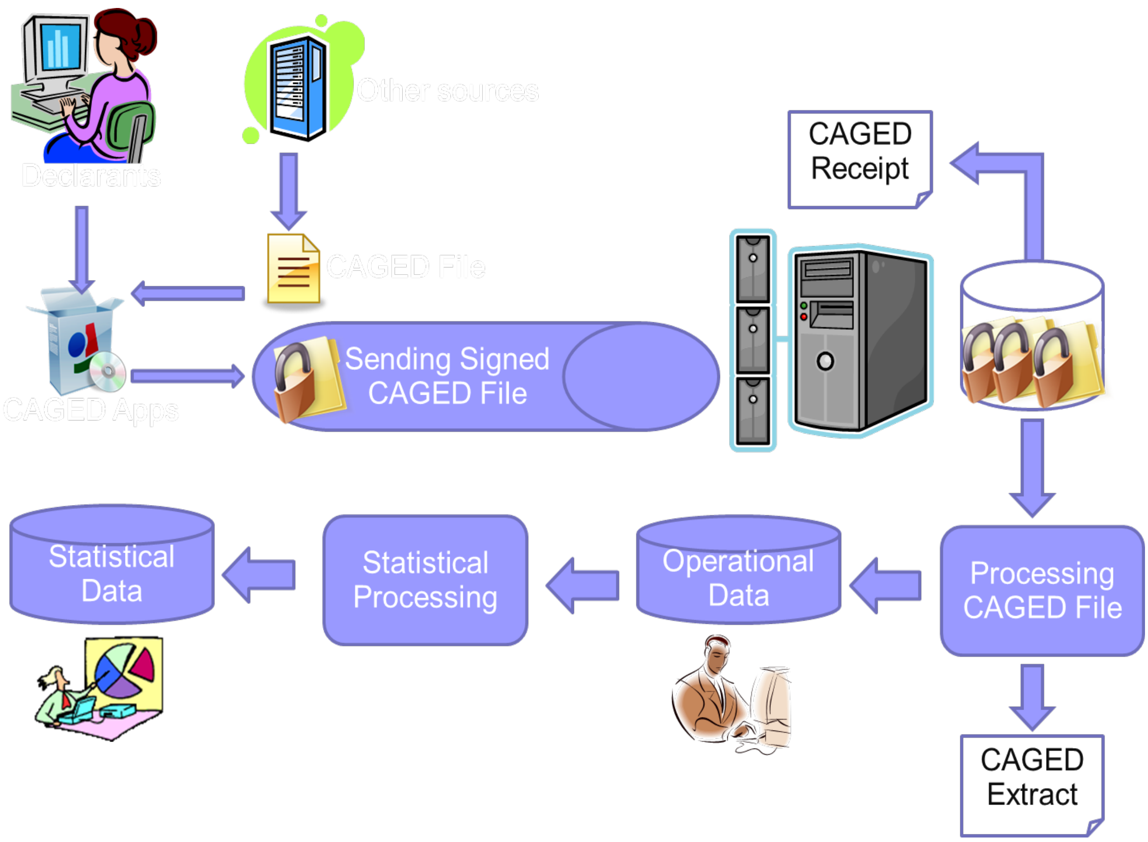


Figure 12 - CAGED Overview

Considering these characteristics, the process is started when the MTE opens the competence of CAGED reception. In this case, the declarants can send their CAGED declaration. The submit process a CAGED declaration is composed of a selection (including filling) of a CAGED file (detect the movements), sending this file and receiving a CAGED Receipt. During the reception, MTE can decide to start the processing of the files that provide operational data for MTE. Depending on timing constraints, such as holidays or some other guidance, MTE staff closes competences of reception. After all checks in CAGED reception and processing, operational staff closes the reception processing and starts statistical processing.

Having in mind that the CAGED submission is an activity that is critical to the success of CAGED process as a whole, we explain the main points that must be considered in its management. The capture operation of a CAGED file can be performed in different manner: (i) Generate CAGED File, that in general is executed through payroll systems of accounting firms which generate a CAGED declaration (in a predefined layout) without MTE analysis and signature; (ii) Generate Analyzed CAGED File, that is executed by declarants through a MTE offline system that generate a CAGED File with analysis and signature; and (iii) Generate Short Analyzed CAGED File, that is performed by declarants to generate and send CAGED declarations with a maximum of 36 movements.

The CAGED Declaration can be generated in two ways: CAGED File without signature of analysis or an Analyzed and Signed CAGED File. The first one can be analyzed through the activity of Generate Analyzed CAGED File or to be analyzed and sent by a service. An analyzed CAGED file, the second one, is sent without additional analysis; in this case, two manners are possible: CAGED File with less than 1500kb can be sent through a service that only verifies MTE signature. CAGED Files with more than 1500kb must be sent throw a desktop tool that optimizes its transmission.

The previews tasks are executed by the declarants that send their CAGED file to MTE. A MTE reception service is available to receive and store these files and triggers the generation of a CAGED Receipt to the declarants.

CAGED is complex system that is hard to be managed. Some qualities attributes must be assured according to its SLA such as availability, response time, processing time, etc; and all these with less human intervention. In this sense, business process models must be able to adequately represent: (i) Mission-critical Activities; (ii) The way the activities are executed and monitored; (iii) Which quality attributes and environmental context must be monitored and to what autonomic features; (iv) How systems needs to be adapted in case of some quality attributes deviation.

3.4 MODELING PHASE

In this section, we present four well-defined abstraction levels of MABUP. Regarding the different levels, we used the nomenclature presented in (ABPMP, 2009), that relies on the use of the concepts related to the hierarchy of a business process as a decomposition of activities and task as shown in **Figure 13**.

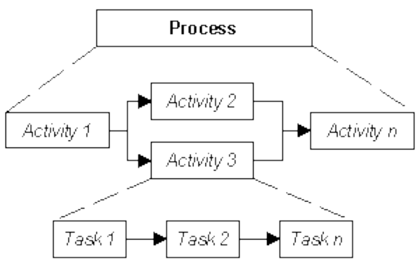


Figure 13 - Process Decomposition

Considering our approach, we argue that organizational level represents the process decomposition in terms of activities and tasks. In turn, in the technological and operational levels we use the term task once this level considers critical activities decompositions and represents atomic actions that must be treat appropriately according to its function. Service level represents the system support to these tasks.

Generally, business analysts model organizational level once they initially organize business knowledge without consider how artifacts must be processed considering technological issues. Technological level, in its turn, generally is modeled by requirements engineering once they are able to understand the correlation between the critical business activity and their refinement in different tasks according the adopted technological solution. What to change in operational environment according with different indicators is defined to ICT professionals, in this sense, they usually collaborate with operational level of the autonomic business process model. Finally, service level is composed of the services that support both monitored tasks and operational tasks, all the information of these services generally is provided by software engineering professionals. In **Figure 14** we present the four abstraction levels and how they are related.

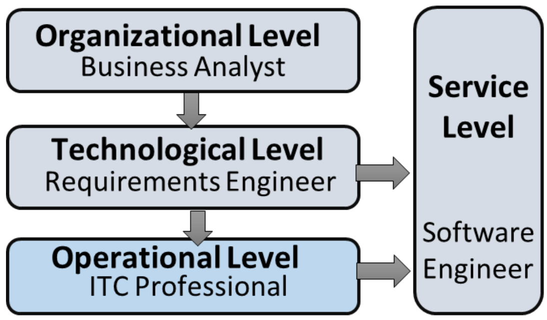


Figure 14 - Multi Level Business Process Modeling

3.4.1 Organizational Level Model

Because the failure of complex, mission-critical systems—such as certain process control and manufacturing operations—can be exceptionally costly to an organization, appropriate management control of these systems is imperative. We argue that activities that must be treated in an autonomic manner are critical to the business and generally a SLA involved.

In this sense, it is necessary to define the object to be adapted that involve the layer, artifact and granularity. For this reason, an organizational level must have a well-defined granularity and supports the modularization of autonomic business process.

Modularization techniques in this level guide the expression of activities that are immune to technological changes.

In order to understand the concept of activity in this level, it is important to emphasize that **entities** represent things within the organization that are of interested in a specific modeling context. Each unit of behavior that an entity plays is abstracted as a role. Thus, to assume that activities are modularized, we have to consider some criteria:

a) **Activity as a unit of functionality:** results from the collaboration between a set of roles;

b) **Processing unit:** involves providing new meaning of information. Similar to the collaboration defined on the classic input-process-output process pattern where the activity transforms inputs into outputs and is performed by an actor playing some role;

c) **Collaborative behavior**: all interactions between entities are always mediated by roles, which abstract their specific collaborative behavior.

In the example presented in **Figure 15**, it is possible to identify how criteria **a, b** and **c** are adopted. The example starts when an entity "E1" is processed into task "Transform" by role "Operator", as output is produced an entity "E2". The E2 is then processed into task "Control Quality" that provides the same entity "E2" to be further processed in task "Transform" by role "Operator" producing entity "E3", which in its turn is again processed into task "Control Quality". Applying criteria **a, b** and **c;** an activity should provide new information to another role. In this sense, entity E1 should be processed to result in another one, in this case, E2 and E3. The refinement of this activity presents how entities are transformed considering different existing tasks. Then, an activity can be interpreted as equation 3.1.

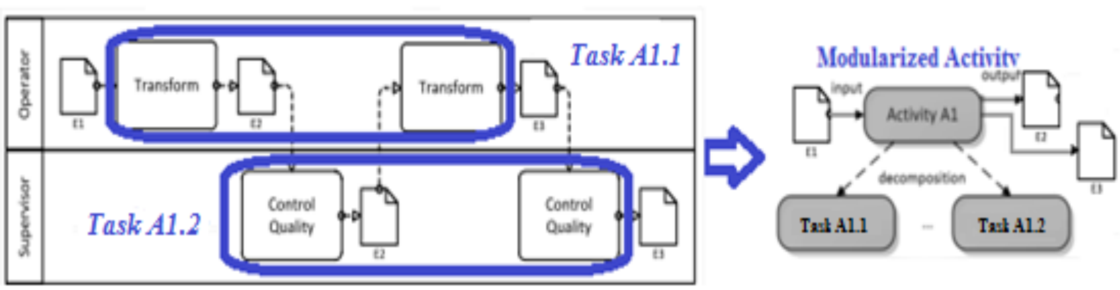


Figure 15 - Modularizing activity according MABUP criteria (a, b and c)

$$Activity = f(a, b, c) \quad (3.1)$$

In this level, we introduce the concept of **critical activity** that is a special kind of activity that is critical to organizational goals and must me refined and treated in autonomic manner to reduce problems that can affects the SLA. Generally, problems that affect critical activities causes serious financial lost, for this reason, this level is modeled usually by business analysts. **Figure 16** presents an example of the concept of critical activity used in our approach showing the data flow of entities E2 and E3 and can be defined as follow:

$$\forall x \in CA, \exists w \in OrgL \mid x = f(w) \quad (3.2)$$

where: OrgL is the Organizational Level; CA is Critical Activity Elements, ie. OrgL has at least one element associated to a Critical Activity.

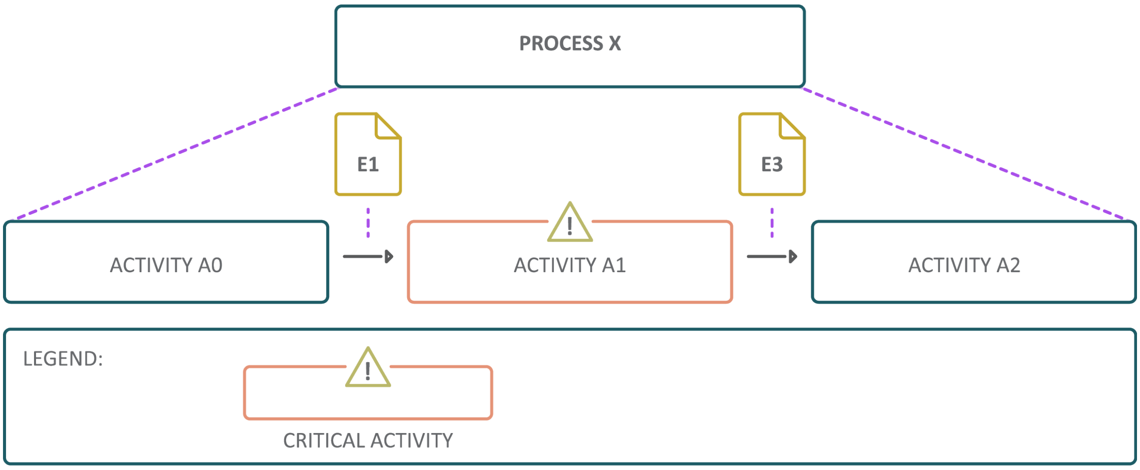


Figure 16 - Critical activity of MABUP

Regarding our running example, **Figure 17** presents the Organizational Level of Business Process of CAGED expressed in BPMN 2.0, where we highlight the critical activity "**Declarant submits declaration**". It is an activity that has as input all CAGED movements and as an output the receipt. Note that it is critical to the success of CAGED process.

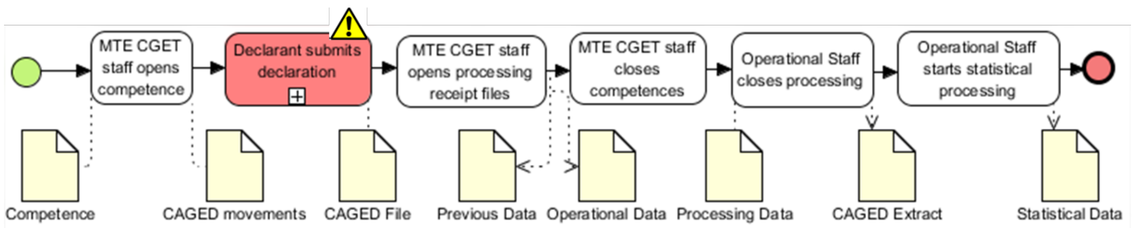


Figure 17 - Organizational level of CAGED

In the sequel, we will define technological level, which describes how a critical activity must be refined to represent autonomic features in order to attend the SLA required.

3.4.2 Technological Level Model

Technological Level represents the decomposition of a critical activity processing to indicate how the activity treat the entries considering business entity, the system, architecture, etc. There are important aspects that can affect software adaptation, such as:

a) **Present different alternatives to perform an activity:** Some activities can be executed in different ways in a company. It is important to identify these differences to analyze if they require special ways to be managed (eg. An activity 'Make financing payment' (critical activity – organizational level) can be done in different ways as make account debit, invoice or personally (tasks – technological level).

b) **Indicate External dependences:** External dependences are important to be expressed as they can lead to interoperability problems. In that sense, interoperability demands human intervention coordination related to the efforts to ensure performance, scalability, correctness, or reliability of applications in the presence of concurrency and failures [2]. We consider that this calls for autonomic characteristics such as optimization, healing and protection. Furthermore, when a system relies on an external service, some kind of service level agreement (SLA) is required (eg. 'Make account debit' accesses an external bank service).

In this level, MABUP introduces the concept of **monitored task**, a special kind of task that must be monitored to assure autonomic capabilities, i.e. Self-Configuration (C); Self-Healing (H); Self-Optimization (O); Self-Protect (P). The choice of monitored task usually consider SLA document that is a good source of information as it presents the quality attributes and their respectively values that must be assured (eg. 'Make account debit' should guarantee self-healing and self-protection characteristics).

In **Figure 18** we present the aspects related to the refinement of a Critical Activity and the Monitored Tasks according MABUP. Technological level includes the following description:

$$\forall y \in MT, \exists w \in TecL \mid y = f(w) \quad (3.3)$$

where: TecL is the Technological Level; MT is Monitored Task, i.e. TecL has at least one Monitored Task.

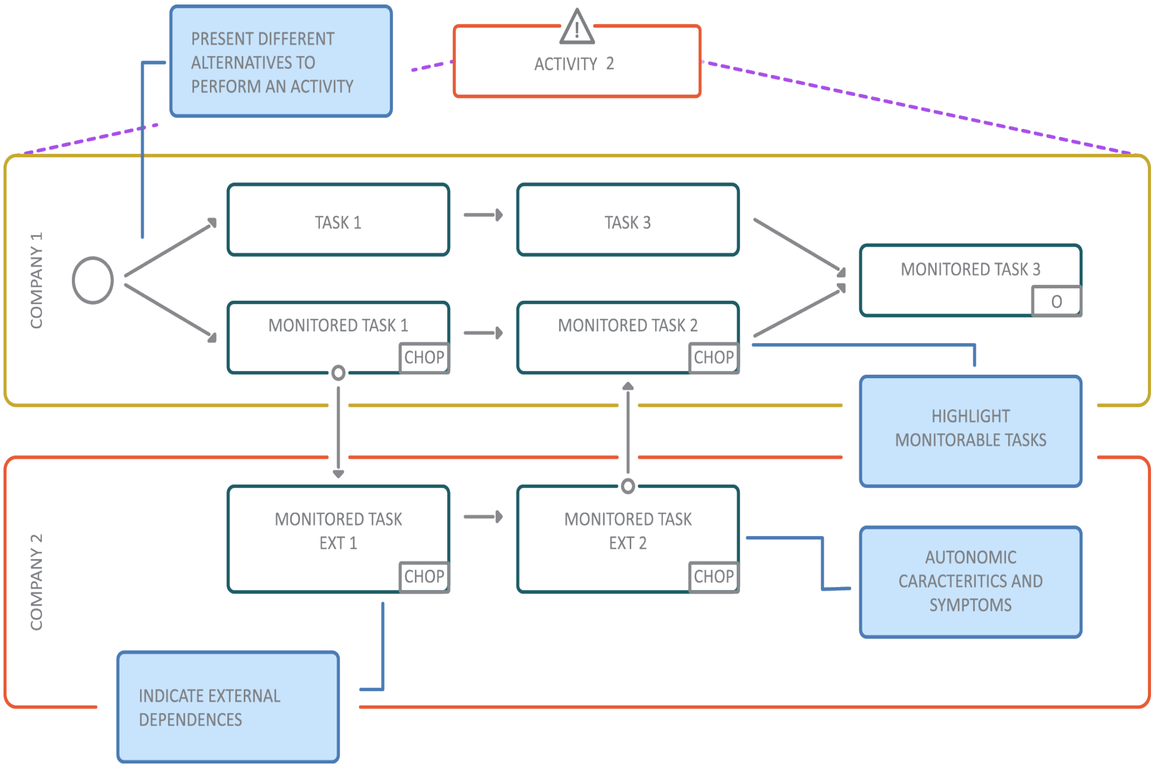


Figure 18 - Monitored Task of MABUP

In our example, the generation of file can be performed in three different manners: (i) Generate CAGED File; (ii) Generate Analyzed CAGED File; and (iii) Generate Short Analyzed CAGED File. If one of these activities became unavailable, another alternative can be executed to guarantee the system operability until all processes return to an optimum state.

Figure 19 indicates that the "Receive CAGED Receipt" task is impacted if the "Receive CAGED File" task becomes unavailable or has a poor performance. For example, the "Generate analyzed CAGED File" task is executed outside the MTE domain by an offline desktop tool that is not expressed in SLA as a monitorable module. In our example the monitored tasks, such as "Receive CAGED File" are depicted in gray. In the running example, we only consider the self-optimization (O) principle to all monitorable tasks. In this sense, if some performance indicator deteriorates, the system optimizes the operational environment resource to return the system to an optimum state.

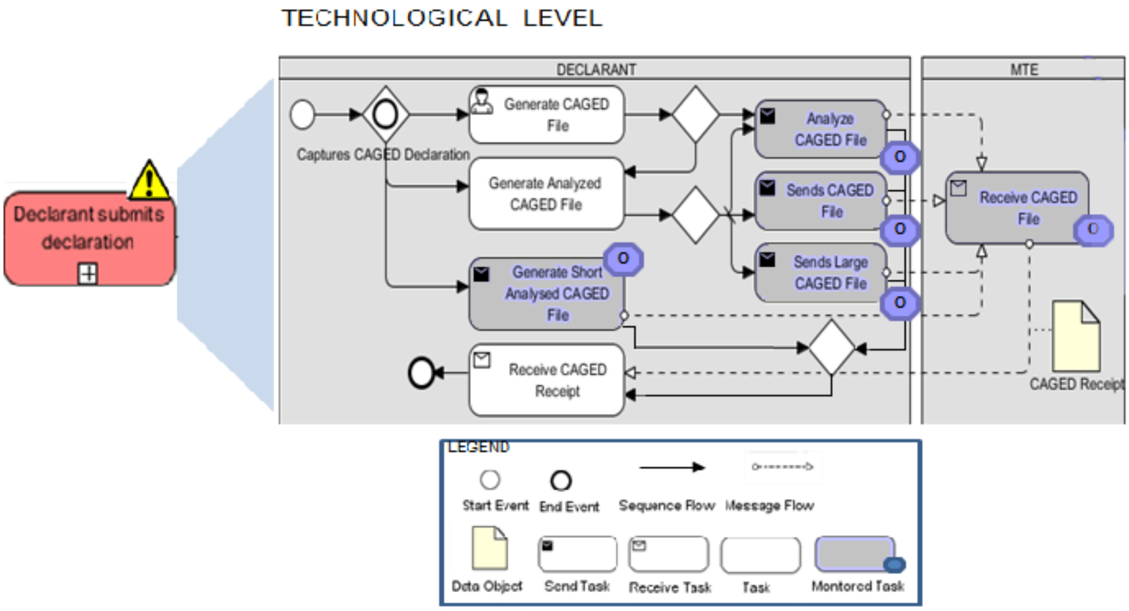


Figure 19 - Technological Level of the Critical Event “Declarant submits declaration”

In next section, we will present operational level, which presents all tasks that could be actuated and in which cases if some deviation in quality indicators occurs.

3.4.3 Operational Level Model

Operational level indicates the operational knowledge required to manage the process. Business analysts define the contextualization of the monitored task using their knowledge about the business domain to identify information that can affect its execution and together with ICT professionals express the operational knowledge to manage it. The information is codified in terms of contexts, context variability and operational tasks.

The variability analysis is focused on context variability. Both variants and variation points are related with to contextual information. They indicate how the contexts variation can affect the autonomic actions. Operational Task expresses autonomic actions in the systems to assure the optimal state of the system and express a refinement of the autonomic part of a Monitored Task.

Considering the SLA of a Critical Activity, the business analysts define the tasks that must be monitored to meet the non-functional requirements agreed. These indicators are verified to analyze if the values obtain from the services that support these tasks presented any deviations. In order to express different variations on NFR deviations, we defined the Variation Point concept in our MABUP approach. Given a Variation Point that represents a NFR deviation, it is important to model the deviation variant. The concept of Variant is directly related to the situation with an expected value of NFR and environment contexts involved.

According to the Variant occurred during the monitored task execution, an operational task should be trigged to adjust operational environment resources. Thus, the operational level presents the new interconnected concepts in ABP models: variation point, variants, operational task and can be defined as equation 3.4:

$$OpeL = \{VP, Var, Ctx, OT, Rel\{VP \times (Var, Ctx) \times OT\}\} \qquad (3.4)$$

where: OpeL is the Operational Level; VP is Variation Point; Var is Variant, Ctx is Context; OT is Operational Task and Rel expresses the relationship between them. OpeL is a conjunct of VP, Var, Ctx and OT where VP is composed by Var and Var is associated with Ctx and OT.

In our running example, we have defined Self-Optimization (O) as the desired autonomic principle. As previously noted, it is related to the efficiency NFR, which in turn can be decomposed into others characteristics such as Response Time and Space Utilization. In this example, we deal only with Response Time attribute.

For instance, a variation point of the Receive CAGED File (see **Table 6**) task is associated with the response time. This variation point is a contextualization for the alternative ways how the adaptation actions are selected to respond to environment changes. The alternatives are represented as variants that are associated with the variation point. In the example, the variants are actions to control or regulate the response time deviation of the CAGED file reception. The tasks are associated with contexts described in expressions that activate the presence of a variant in the variation point.

Given that in our running CAGED there is a deadline for declaration submission, it is important to measure the daily reception. As observed in **Figure 20**, the reception rate peaks during the first seven days of the month. This trend must be considered to allow the selection of a correct adaptation. Trend analysis methods (VEDAM; VENKATASUBRAMANIAN, 1997), for example, can help business analysts to predict future outcomes tracking variances in historical results. In this sense, the contexts verify the trend of reception in the beginning of the month.

Table 6 - Contextualization of the Monitored Task "Receive CAGED File“

Monitored Task	Contextualization			
	Variation Point	**Variants**	**Context**	**Quality**
Receive CAGED File	Response Time Deviation	Increase Reception Rate	C1: ReceptionTrendIsOK= true and LastThreeCycleIncreasing= true	Response Time >= 220ms
		Decrease Reception Rate	C2:ReceptionTrendIsOK= true and LastThreeCycleIncreasing= false	Response Time <= 110ms
		Deviation Reception Rate	C3:ReceptionTrendIsOK= false and LastThreeCycleIncreasing= true	Response Time >= 220ms

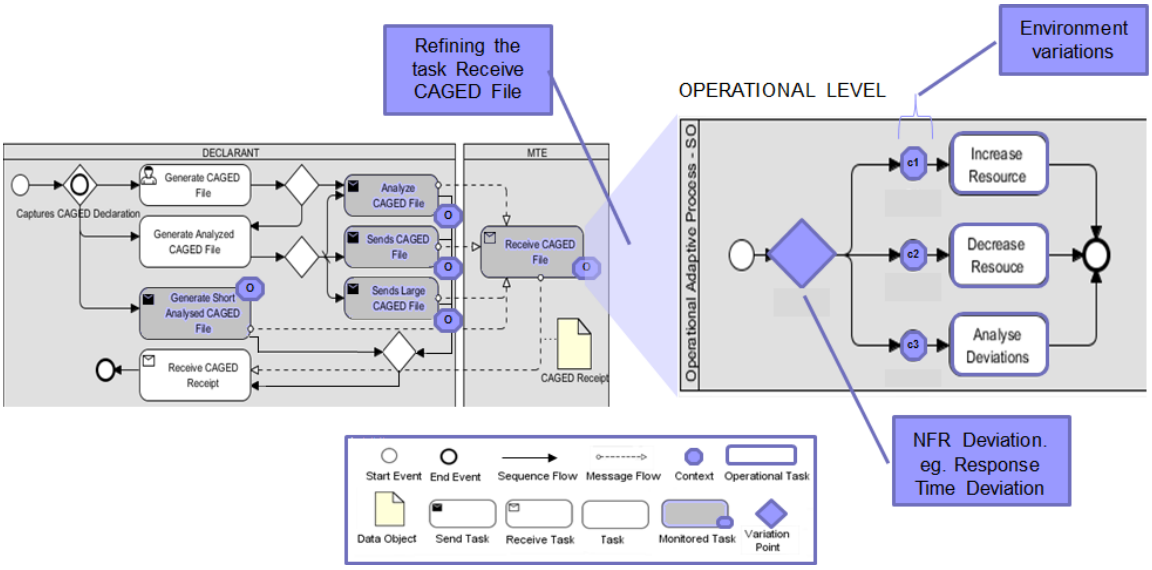


Figure 20 - Operational Level of the Monitored Task “Receive CAGED File”

In order to treat Response Time deviations that may be related to the performance of the “Receive CAGED File” activity, we defined in **Figure 20** (Operational Level) three different tasks (operationalization): (i) Increase resource; (ii) Decrease Resource; and (iii) Analyze deviation.

The three contexts (C1, C2 and C3) present in **Figure 21 -** Contexts of the Monitored Task "Receive CAGED File“ (Operational Level) are defined in **Table 6**.

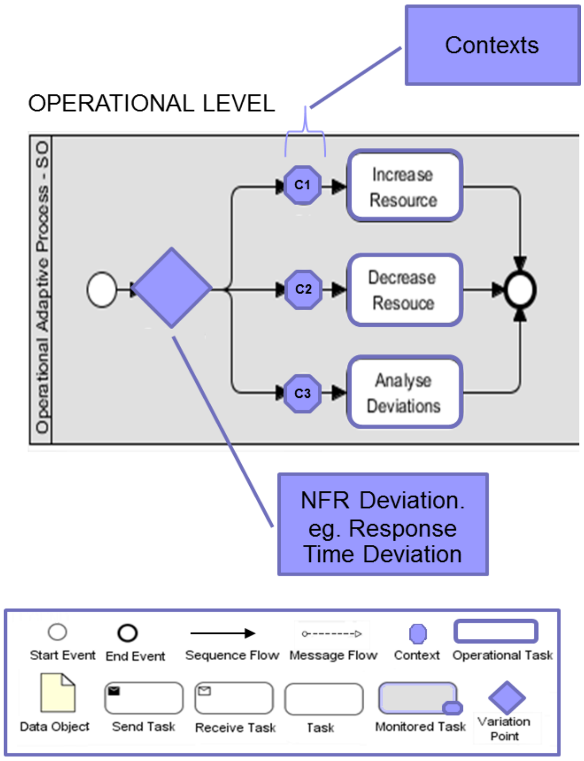


Figure 21 - Contexts of the Variation Point "Response Time Deviations"

Considering the levels already presented, in the sequel we will describe the service level that represent the services that support both concepts of monitored task and operational task.

3.4.4 Service Level Model

Both monitored and operational tasks should be linked to system services. In a service-based mission-critical system, adaptation is an activity with the objective of delivering an acceptable/guaranteed service based on SLA.

One of the key components in SLA is SLO (Service Level Objective) which specifies QoS (Quality of Service) metrics governing service delivery. Each SLA may include several SLOs, each corresponding to a single QoS parameter related to quality factors.

In the Service Level, the services linked with monitored tasks are called monitored services that should be checked according to the parameters presented in the SLO. Whereas the services linked to operational tasks are named as operational services that have the objective of returning the system to an optimal state in an autonomic manner. The concepts used in this level are related as follow:

$$MonitoredService \subseteq Service \qquad (3.5)$$

$$OperationalService \subseteq Service \qquad (3.6)$$

$$MonitoredService \cap OperationalService = \{\} \qquad (3.7)$$

$$MonitoredService \cup OperationalService \subseteq Service \qquad (3.8)$$

That is, the equation 3.5 and 3.6 show that *MonitoredService* and *OperationalService* are a sub-set of *Service* and equation 3.7 and 3.8 show that they are mutually exclusive, i.e. a disjoint and incomplete inheritance of *Service*. Furthermore, other services may exist to support the whole process model.

For example, the "Receive CAGED File" activity is linked to "Service5" that should assure a response time of 190 ms.

In next section, we will present the meta-model of our approach where we define how the aforementioned concepts are related and the constraints are involved to its use.

3.5 META-MODEL OF OUR APPROACH

A metamodel provides an abstract syntax to distinguish between valid and invalid models. That is, a metamodel specifies modeling constructors, their relationships and their well-formedness rules. Hence it contributes to turn the models less ambiguous and more correct. Moreover, a metamodel also can serve as a straightforward option for interchanging the metadata with other tools (KELLY; TOLVANEN, 2008)(MORENO; ROMERO; VALLECILLO, 2008). In this context, in Figure 10 we have proposed the Multi-Level Autonomic Business Process Metamodel (MABUPMM), which specifies the modeling elements of our approach and can be used to provide facilities for interchanging metadata in autonomic business process environments. MABUPMM considers the expression of self-aware and context-aware concepts using the Critical Activities, Monitored Task, Operational Task, Context Variability and NRF. Modularization and separation of concerns were explored by the use of four defined levels.

MABUPMM has two enumerations, which cover one of the possible valid values for an attribute. The AutonomicPrinciple enumeration is a powerset representing all combinations (except the empty set) among the following autonomic principle: Self-Configuration (C), Self-Healing (H), Self-Optimization (O) and Self-Protecting (P). In turn, the Operator enumeration represents the basic logical operators (including "NONE" option) that may be used to define a logical expression.

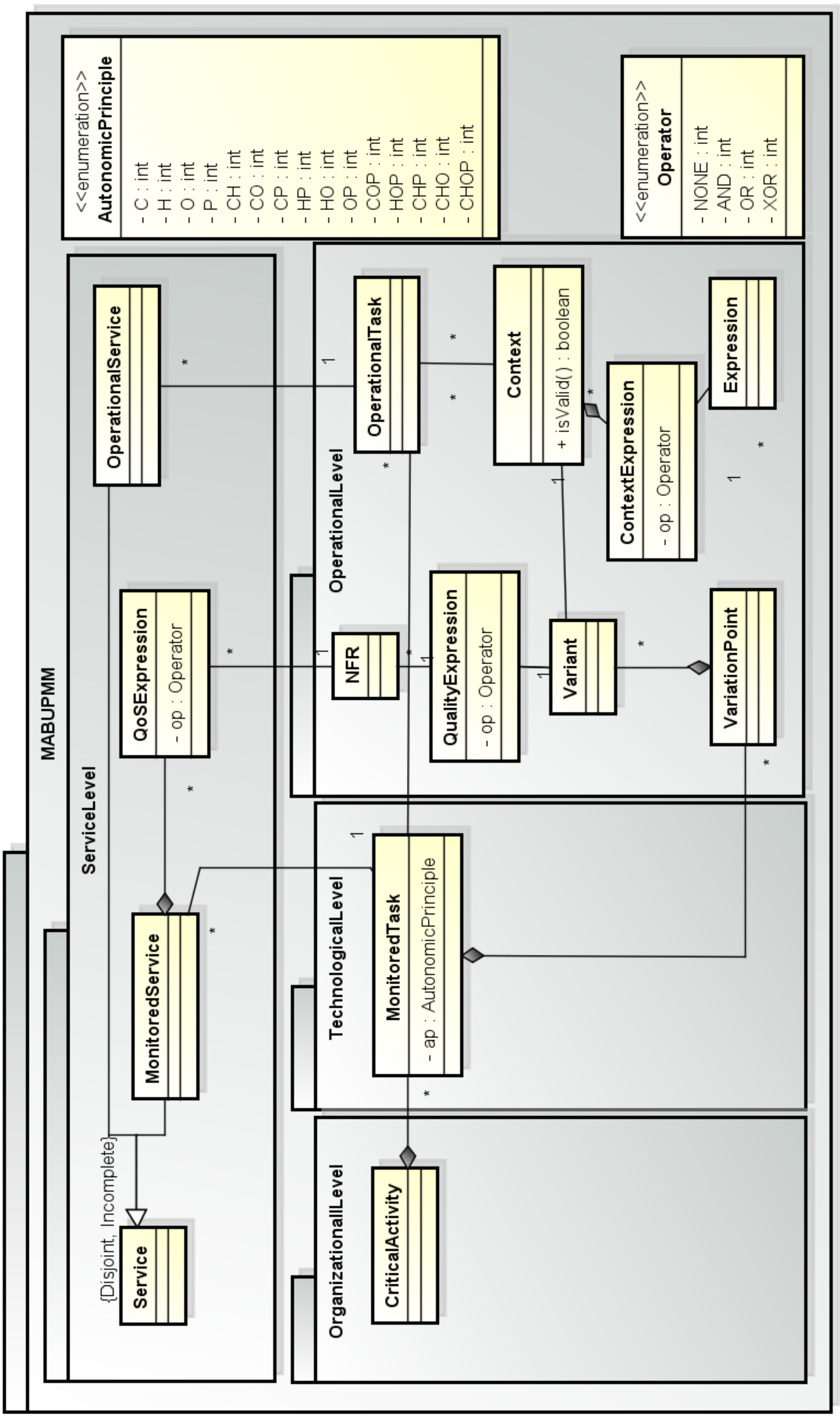


Figure 22 - Multi-Level Autonomic Business Process Metamodel (MABUPMM)

Besides the enumerations, the MABUPMM has fourteen classes and three interfaces, which are organized by its respective packages. The **Table 7** resumes the meaning of these elements (more detail may be seen in Section 3.4).

Table 7 - MABUPMM Concepts

Concept	***Description***
Critical Activity	Mission-critical activity that must be refined to provide information about its behavior and indicate the tasks that must be monitored.
Monitored Task	Task that must be monitored according autonomic principles.
Variation Point	Contextual information related to environment variation that affects the system, i.e. contextualization for alternative ways of how the adaptation actions are selected to respond to changes.
Variant	Alternative associated with the variation point.
NFR	Quality attribute that must be assured by the system
Quality Expression	Expression related to the quality attributes metrics that must be checked in the variant.
Context	Set of environment situations that occur and affect the business process execution.
Expression	Atomic sentence that represents an environment situation and its comparative value.
Context Expression	Sentence with the Expression that compose a Context.
Operational Task	Task that express autonomic action in the systems to assure the optimal state of the system.
Service	System operation that executes a process (ie. Process, activity or task) independent if it is autonomic or not.
Monitored Service	Service that has a specific SLO to be assured.
Operational Service	Service that executes operational task to recover the system to an optimal state.
QoS Expression	Expression related to the metric of quality attributes contained in the SLO and provided by service monitoring

MonitoredService and *OperationalService* are a disjoint and incomplete inheritance of Service because they are mutually exclusive and other services may exist (but these one are out of scope of MABUPMM). Moreover, all metaclasses have the attributes id and name, which correspond an artificial identifier and a descriptive label, respectively (these attributes are not shown in **Figure 21** in order not to pollute it). With respect to the *ap* and *op* attributes, they correspond to a single value of the *AutonomicPrinciple* or

Operator enumerations. The *isValid*() operation checks whether some *ContextExpression* of a *Context* is inconsistence (eg. ReceptionTrendIsOK= false AND ReceptionTrendIsOK= true).

In **Figure 21**, the metaclasses are associated as follow: a *CriticalActivity* is a composition of many *MonitoredTask*, which is a composition of many *VariationPoint* and many *AutonomicTask*. Moreover, a *MonitoredTask* is associated with many *MonitoredService*. In turn, a VariationPoint is a composition of many Variant that is a composition of many *QualityExpression* and many *Context*. The former metaclasse (*QualityExpression*) is associated with a single *NFR* that may be reused in many *QualityExpression*. On the other hand, the latter metaclasse (*Context*) is a composition of many *ContextExpression*, which is associated with an Expression that may be reused in many *ContexExpression*. Then, a *MonitoredService* is associated with a single *MonitoredTask* and is a composition of many *QoSExpression*, which is associated with a single *NFR* that may be reused in many *QoSExpression*. An *OperationalService* is associated with a single *OperationalTask*, which may be associated with many *OperationalService*.

We examined the extension of two business process modeling languages, ie. Communicative Event Diagram (CED) and BPMN, to incorporate MABUP concepts. The description of these extensions are detailed in **Appendix A and B.**

3.6 MANAGEMENT PHASE

In this section, we present our MAPE mechanism that assumes the use of the multi-level autonomic business process model previously described. We then define the behavior of four MAPE modules, ie. monitor, analyzer, planner and executor, to use the different information represented in the model and guide the instrumentation of the system.

3.6.1 Monitor

The monitoring component collects indicators provided by the system from time to time (cycle). The monitor checks both context information and measures associated to NFRs related to the system. In our approach, we assume the use of context sensors, which are computational entities that provide raw data about the operational environment presented in the instrumented BPM; and service quality attributes that must be assured. The monitor module checks the processes at a well-defined level of modularity

(Technological Level) and affects services-oriented systems with coarse grained adaptation at run-time.

As shown in **Figure 23**, the monitor module is divided in Context Monitor, QoS Monitor and Monitor Engine. The context monitor reads and processes the contextual data provided by the instrumented system. The QoS Monitor reads and processes the log database of all monitored services that support all monitored tasks and their respective metrics according to the SLO. The Monitor Engine processes the information provided by Context Monitor and QoS Monitor, collects the relevant data according MABUP model and passes it to the diagnostic component for analysis.

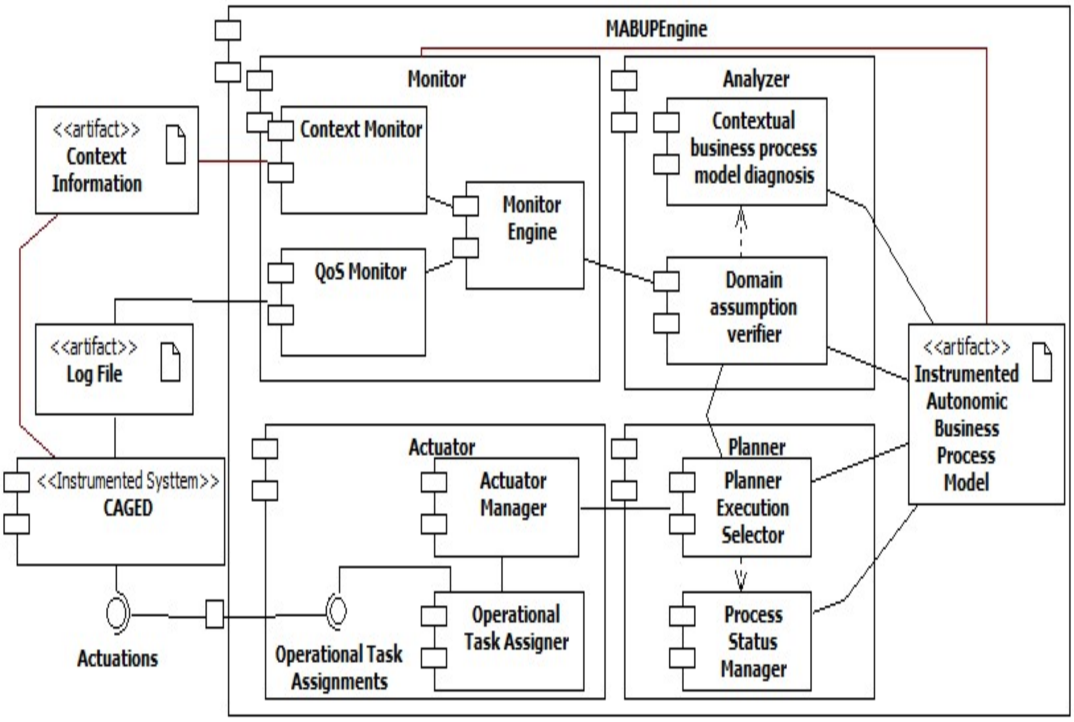


Figure 23 - MABUP Engine

Algorithm 1 defines how the monitor process should work, ie. it obtains the MABUP model, treats the monitored tasks, collects the metrics of each one and returns the data to be used by analyzer module.

Algorithm 1. Monitor algorithm

monitor(MABUPModel model)
1. MonitoredTaskValue[] mtvs
2. VariationPointValue vpv
3. VariationPointValue[] vpvs
4. **for each** tl ϵ getTechnologicalLevels(model)
5. **do for each** mt_i ϵ getMonitoredTasks(tl)
6. MonitoredService ms ← getMonitoredService(mt)
7. OperationalLevel op ← getOperationalLevel(mt)
8. **do for each** vp_j ϵ getVariationPoints(op)
9. vpv ← getVariationPointValues(mt,ms,vp)
10. vpvs[j] ← vpv
11. **end for**
12. mtvs[i] ← getMonitoredTaskValues(mt, vpvs)
13. **end for**
14. **end for**
15. **return** mtvs

The *getMonitoredTaskValues* method (line 12) localizes, though QoS Monitor, the metrics of the monitored service expressed in the Service Level based on the identification of the monitored task. It calculates the average of the metrics obtained in a cycle by all requested services that support the monitored task as follow:

$$f(s) = \frac{1}{n}\sum_{i=1}^{n} QoS(s_i)$$

where: f(s) the function to verify the QoS of a service s; n is the number of requested services in a cycle; QoS(s_i) is the metric obtained for a requested service.

With this information, an analyzer module verifies the threshold defined in business process model. In next section, we will describe how MABUP approach define analyzer module.

3.6.2 Analyzer

In our approach, this component has the objective to evaluate the metrics obtained by the monitor. Considering the measures obtained by the monitor, the diagnostic component checks the contexts expressed in the Business Process Model. In case of some deviation, the planner (section 3.6.3) selects the appropriate operational task that represents autonomic interventions in the system.

Considering the management of business processes, it is important to assure the service level agreement related to these processes. In this sense, the analyzer module checks the monitored metrics in the business process model and in case of deviation above a predefined threshold the module checks the operational contexts that affect the process and the variants related to these contexts. This information trigger the activation of planner module. The analyzer module is divided in two components: Domain Assumption Verifier and Contextual Business Process Model Diagnosis.

The "Domain Assumption Verifier" component identifies all quality attributes related to the monitored task and in case of metric deviations, the "Contextual Business Process Model Diagnosis" component analyzes the contexts values. **Algorithm 2** is a simplified version of this process. According to lines 2-4 the "Domain Assumption Verifier" component reads all monitored task values obtained by the monitor module and verifies the QoS deviation. The isQoSDeviated method (line 3) reads the Service Level, gets the expected metrics and compare them against the obtained values. In the "Contextual Business Process Model Diagnosis" component, the getDeviatedVariant method (line 5) checks all Variation Points and evaluates the context and quality expected in each Variant. The Analyzer module returns all the deviated variants.

Algorithm 2. Diagnosis algorithm

---

analyze(MonitoredTaskValue[] mtvs)
1. Variant[] deviatedVars
2. **for each** mtv ϵ mtvs
3. **do boolean** isQoSOut ← isQoSDeviated(mtv)
4. **if** isQoSOut **then**
5. deviatedVars += getDeviatedVariant(mtv)
6. **end for**
7. **return** deviatedVars

In our running example we had explored the monitored task "Receive CAGED File" that has the Self-Optimization (O) as the desired autonomic principle. The respective operational decomposition presents the Variation Point related to the Response Time attribute. In this sense, we explored three scenarios according the data provided in **Figure 21** that are related to the tree variants presented in this Variation Point:

Scenario 1: In the beginning of the month the reception rate is increasing time-to-time and this is an expected change in the environment according to the CAGED characteristics. The system provided the following information: ReceptionTrendIsOK=true; LastThreeCycleIncreasing=true (Contextual); and ResponsteTime=382ms (QoS);

Scenario 2: After the deadline to send the CAGED File, the competence (month of the CAGED movements) is closed to receive files of previous month and opens to receive the next competence. As result, reception rate decreases time-to-time. Considering this, the system provided the following information: ReceptionTrendIsOK=true; LastThreeCycleDecreasing=true (Contextual); and ResponsteTime=80ms (QoS);

Scenario 3: The reception is out of the reception peak and the reception rate is increasing, as shown in **Figure 24** (10/06/2011). In this case, the system must evaluate the deviation but not increase the resource. Considering this, the system provided the following information: ReceptionTrendIsOK=false; LastThreeCycleIncreasing=true (Contextual); and ResponsteTime=457ms (QoS);

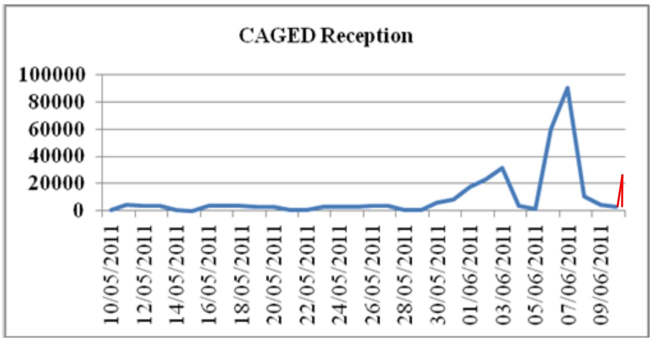


Figure 24 - CAGED Reception Rate

Considering scenario 1, 2 and 3; in different moments the analyzer module would return respectively the variants: Increase Reception Rate, Decrease Reception Rate and Deviation Reception Rate. It is important to emphasize that even if a quality attribute is out of the defined value, the deviated variants will only be returned only if there is contexts that enables it. For example if the system returns a response time of 350 ms in the execution of the task “Receive CAGED File” and there is no context that influences in this value, the autonomic engine will not return deviations.

3.6.3 Planner

The main objective of planner module is planning the interventions that will be executed in the system according the information provided by the analyzer module. As shown in **Figure 23**, the planner module has two components: The Planner Execution Selector and the Process Status Manager. The component Planner Execution Selector reads all deviated variants obtained through analyze module and then obtain, from the Operational Level, the Operational Tasks related to these variants. The status of each selected Operational Task is managed through the Process Status Manager component to assess if this task can be executed. The status can indicate if there are other executions of the same Operational Task in the moment (in_progress) or if there are some previous plans that have the same operation and have not stated yet (uncommitted). In this case, the Operational Task does not enter in the plan that will be executed.

Regarding our running example, considering Scenario 1 (defined in section 4.2.2), the planner module returns the “Increase Resource” Operational Task. While in Scenario 2 the “Decrease Resource” Operational Task is returned. Finally, in the Scenario 3, the module returns “Analyze Deviations”.

3.6.4 Executor

The actuator module has the objective of actuating in the instrumented system. For this reason, the module has two components as presented in **Figure 23**: The Actuator Manager and the Operational Task Assigner.

According the Operational Task returned by the planner module, the Actuator Manager accesses the Service Level to obtain the Operational Service with the parameters that it needs to be executed. The Operational Task Assigner access the instrumented system to finally perform the adaptation.

Considering our running example, if there is a plan to perform the operational task “Increase Resource”, the Actuator Manager component will access the specification of the service named “OP1”. This specification has the name of the service that is: *expandReceptionMem(ResourceType rt, ResourceValue rv, SystemModule sm)* where *ResourceType* is the type of resource that the analyst wants to increase, *ResourceValue* is the percentage of growth and the *SystemModule* is the identification of the sub-system.

These values are predefined as in this case as: rt=" memory"; rv=0.1; and sm="cag.Reception". It corresponds to 10% of increase of memory resource.

3.7 DISCUSSION

The modularization helps to treat the complexity of large-size models (REIJERS, 2008). Once the management of business process models can be realized at different levels of granularity, there is a tradeoff between monitoring granularity and diagnostic precision. The finest level of monitoring granularity is at the functional level where all leaf level tasks are monitored. The disadvantage of fine-grained level monitoring is high monitoring and diagnostic overhead. Coarser levels of granularity only monitor higher-level business goals. In these cases, less complete and high-level monitoring data is generated, leading to multiple competing diagnoses. Both monitoring and diagnostic overheads are lower. The disadvantage is that if requirements denials are found, multiple diagnoses are returned, each pinpointing possible failures (WANG; MYLOPOULOS, 2009).

We argued that the quality attribute is not sufficient to represent deviation in autonomic scenarios. It is also necessary to express the context variability of the operational environment. Having this in mind, we advocated for the explicit representation of environment variability through context-awareness and quality attributes in order to properly represent autonomic features.

According to the taxonomy of self-adaptation (SALEHIE; TAHVILDARI, 2009), in order to represent adaptability some concepts must be presented: (i) object to be adapted; (ii) realization issues; (iii) temporal characteristics; and (iv) interaction concerns. Considering this taxonomy, our approach relies on business process based adaptation that has a well-defined level of granularity and affects services-oriented systems with coarse grained adaptation at run-time.

Due to these characteristics, the business analysts must use their knowledge and the information about the business domain to explicitly select the proper adaptations. It is necessary to elicit autonomic business (TERRES; NT; SOUZA, 2010), for example, the authors proposed a method to select business process to autonomic automation. Other possible solution is the use of trend analysis to identify the context expressions and the recurrent problems in their operational environment (VEDAM; VENKATASUBRAMANIAN, 1997).

3.8 Chapter Summary

In this chapter we outlined an approach for managing autonomic business processes following a multi-level strategy. An architecture of autonomic business process models that intends to represent the correct behavior of the systems from the business, technological, operational and service points of view. This proposal has been divided into two parts: the modeling and the management phases. The modeling phase express how to obtain four abstract levels that instrument the business process model with autonomic features; namely, Organizational Level, Technological Level, Operational Level and Service Level. The management phase proposes a MAPE (Monitor-Analyze-Plan-Execute) engine that considers both instrumented ABPM (Autonomic Business Process Model) and the instrumented system obtaining information about the environment context and quality attributes to guide the adaptation. We introduced algorithms to perform the MAPE modules of our approach.

Furthermore, we outlined a meta-model to our approach that presents the relationship of the aforementioned concepts. These concepts were designed so that they can be used in different business process modeling notation.

We have presented an illustration of the feasibility of our approach. We utilized BPMN 2.0 to represent Organizational, Technological, Operational and Service Levels because it is a widely used standard.

In our evaluation, we presented the autonomic feature of self-optimization. More precisely, we detailed the assurance of the characteristic of Response Time. The real example demonstrates the feasibility of the presented conceptual architecture and highlights its applicability to a real-life scenario. Furthermore, the benefits of our approach are manifold and include:

- *Expressiveness of autonomic features in BPM*: In contrast to other approaches that need a knowledge database to express the metrics that affects the adaptation, our approach provides the concepts of critical event, monitored task, context variability and quality attributes expressed in a BPM that guide the self-management at runtime. All these concepts are interconnected to indicate how the metrics impact each autonomic business process.

- *Separation of concerns:* In MABUP model the relationship between the business, technological and operational knowledge are explicitly expressed in different and interconnected abstraction levels of the model.
- *Modularity of ABPM:* We have relied on modularization principles to deal with the overhead in the monitoring phase. The Organizational Level helps to indicate mission-critical activities that must be treated in an autonomic manner. The Technological Level, a refinement of the mission-critical activities, specifies the autonomic tasks that must be monitored.
- *Scalability*: Modularity helps abstracting processes, thus reducing the complexity of ABPM, and increases the scalability of our approach;
- *Understandability*: The different well-defined levels helps to provide understandability of ABPM;
- *Guide the adaptation based on metrics, as context and quality attributes, contained in the BPM*: Considering the knowledge provided in our MABUP model, the management module guides the adaptation according the context and quality attributes that affect the operational tasks.

4 Experimental Evaluation

In order to highlight the gains on the use of MABUP approach, this section succinctly describes evaluation techniques to validate the technique and the obtained results.

4.1 Evaluation Perspectives

The evaluative planning of MABUP was conducted from two perspectives: (i) Simulation on the use of MABUP concepts and (ii) Controlled Experiment, both conducted based on a real scenario. The different perspectives of evaluation led to the detection of problems and results in many ways, making a complete analysis. The evaluation mentioned in this plan is based on the multidimensional evaluation approach presented in Bohme & Reunssner (2008).

Simulation is a powerful approach to predict the impact of a certain BP design on the quality of Information System (IS), so BP design and IS design can be aligned by making adjustments based on the predicted quality impact. In this case, the simulation was used to measure consumed time of operational execution, total cost and workload on using MABUP (management phase).

Controlled Experiment allows conducting well-defined, focused studies, with the potential for statistically significant results (WOHLIN et al., 2000). They allow focusing on specific variables, measures, and the relationships between them. They help formulate hypotheses by forcing a clearly state of the question being studied and allow maximizing the number of questions being asked. Such studies usually result in well-defined dependent and independent variables and well-defined hypotheses. They result in the identification of key variables and good proxies for those variables. They allow measuring the relationships among variables. In this case, the controlled experiment was used to evaluate expressivity of autonomic features, modularity technique, complexity and modeling time on using MABUP (modeling phase).

<u>Experimental Object</u>

We evaluated our approach with a real example (WIERINGA; MORALI, 2012). The example consists of a large system that requires to be managed in an autonomic manner. Specially, we treated the characteristic of self-healing. We examined the

‘*Telecom Service Management’* process that is part of the value chain of a real telecom company.

We modeled our experiment on Academic Signavio Process Editor (available in http://academic.signavio.com) to facilitate the simulation and after exported the model to complete its documentation using Activiti Plataform 5.16.2, that is a light-weight workflow and Business Process Management (BPM) and had an extension of MABUP concepts.

The ‘*Telecom Service Management’* process supports the operation of telecom services contracted by corporative clients to meet their business requirements. These requirements are analyzed by an account manager that formalizes the Telecom Services Contract that predicts a SLA. He also produces a work order (WO) to be implemented. Network provisioning area examines WO, creates the telecom service as specified, activates it and implements operational services that must manage operational environment. After the telecom service activation, subscribers can use it and the telecom company starts to control their access until receive a request to suspend the services contracts.

The activity that controls the telecom service access is critical to the success of telecom service management process. Failure of this activity can cause an exponential financial loss. For this reason, it must be refined and treated in an autonomic manner. The objective of this activity is to receive a request access, to charges, inspect the service access and finalize the request.

Technological level represent the refinement of the ‘*Control Access for Telecom Service”* task. The task must be monitored according to autonomic features. This activity starts when a telecom service request is received, which can be a start, an update or an end request. When accessing the service for the first time, it is necessary to check the access allowance by verifying if the subscriber has access quotes to the requested service. If so, billing area records the use of this quote and mediation area allow the service traffic. Network area, in turn, allows the service to start. There are four predicted services: mobile data, fixed internet, fixed voice long distance and mobile voice long distance. When a quote is being used, its traffic is inspected to request the continuity of the service access by sending an update request for a new quote. Then, the system realizes the online billing to allow or deny access. If online billing fails, the system can perform a hot billing (ie. billing in a short period) that is more costly and susceptible to problems. In any time the subscriber also can request the end of service access. In **Figure 24** the different alternatives indicate what should be monitored.

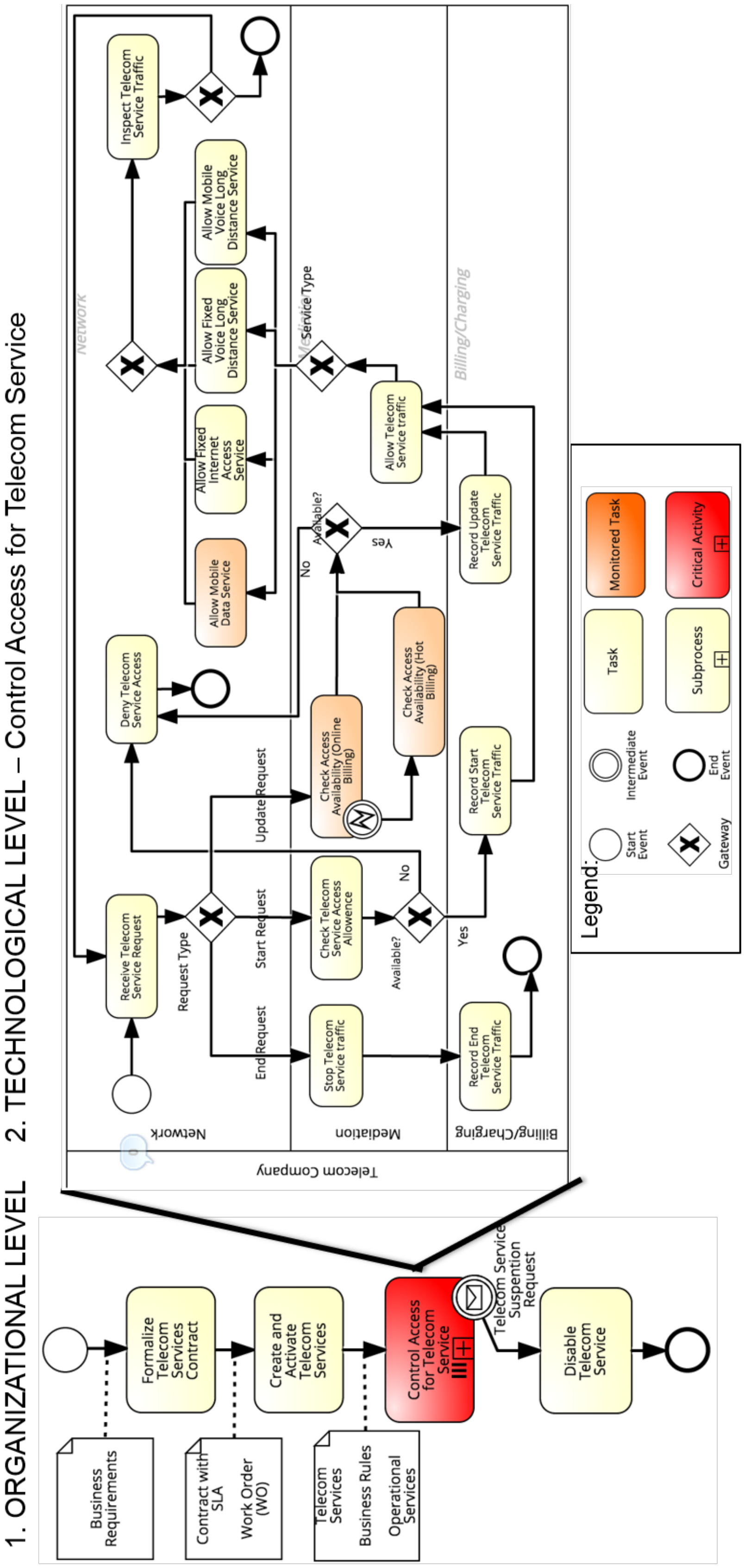


Figure 25 - Organizational Level and Technological Level of MABUP Approach

In our running example, we had explored the "Allow Mobile Data Service" monitored task that has the self-healing (H) as the desired autonomic principle and in its operational decomposition presents the variation point related to the *data loss attribute*. In order to treat data loss deviations that may be related to the *performance* NFR, we defined in operational level three main different operational tasks: (i) Increase 1 db laser power of the transmitter; (ii) Restart transmitter board; and (iii) Restart network gateway. The three contexts (C1, C2 and C3) present in **Figure 26** (Operational Level) are defined in **Table 8**.

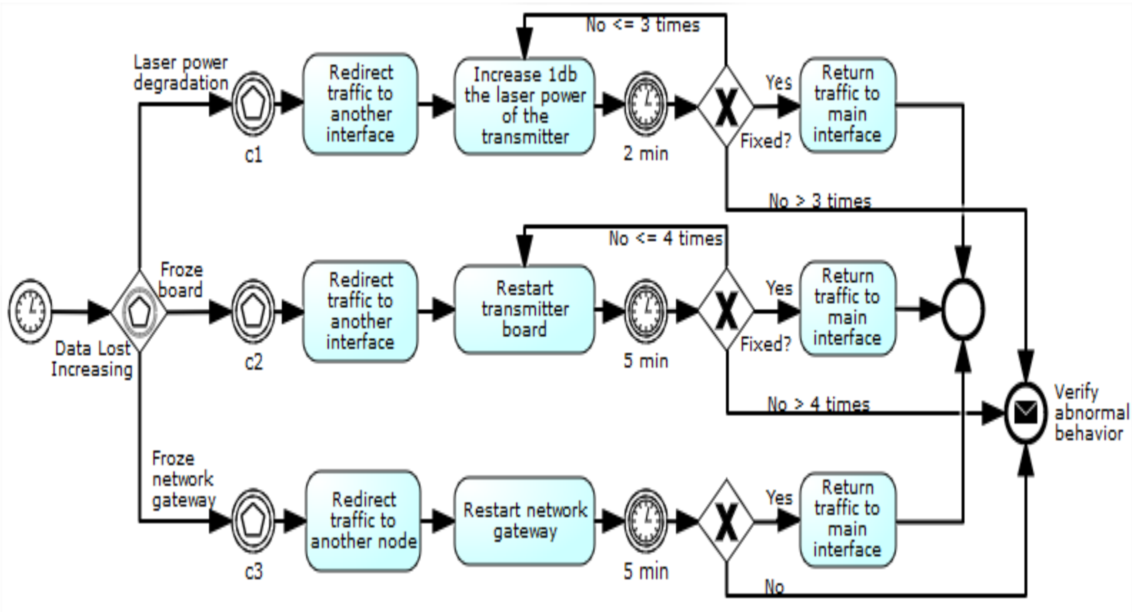


Figure 26 - Operational Level of the task Allow Mobile Data Service

Table 8 - Contextualization of Autonomic Business Process

Task	**Contextualization**			
	Variation Point	***Variants***	***Context***	***Quality***
Allow Mobile Data Service	Data loss increasing	Transmitter laser power degradation	C1: erroIncreasingLastThreeCycles=true and receptionPowerChanged=true	DataLoss <0.001
		Froze board	C2: erroIncreasingLastThreeCycles =true and receptionPowerChanged=false	DataLoss <0.001
		Froze network gateway	C3: erroIncreasingLastThreeCycles =true and receptionPowerChanged=false	DataLoss <0.001and ReponseTime >100 ms

Service Level specifies the services that support the necessary monitored task and operational task. These tasks are specific type that represents an automated task. Such tasks can invoke a service. If necessary technical details to the business process can be added, such as:

<serviceTask id="mobiledata" name=" Allow Mobile Data Service" activiti:class="org.mabup.bpmn.servicetask.AllowMobileData""> …</serviceTask>

After describing all services, the business process must be executed with its support system. Instrumented system should provide quality attributes and context information to guide adaptations. In this sense, we explored one scenario according to the data expressed in **Table 8** that are related to the three variants presented in the variation point.

Scenario: The corporative client contracted an availability of 99.9%. The monitor component collects indicators provided by the system and the analyzer module checks that data loss is increasing time-to-time which is an expected change in the environment according to the Telecom Service Management characteristics. The system provided the following context and quality attribute:

erroIncreasingLastThreeCycles=true and receptionPowerChanged=true (Contextual); and DataLoss = 0.03 (QoS).

Considering this information, the planner module indicates that the system should trigger the following tasks: 'Redirect traffic to another interface', 'Increase 1 db the laser power of the transmitter' and verify if the solution fixed the problem. Then, the Actuator module triggers the operational services to adapt system environment.

In the sequel, we present the simulation of telecom service management process to evaluate its benefits in a real life scenario.

4.2 SIMULATION OF MABUP MODEL

In order to present another evidence of the benefits on the application of MABUP approach, in this chapter we describe minutely the simulation of a MABUP model comparing with real world results to evaluate its adoption.

Böhme and Reussner (BOHME; REUNSSNER, 2008) introduce three types of validations for prediction methods. Type I (Feasibility) studies validate the accuracy of a prediction method by comparing prediction results to measurements from reality or results of another method. Type II (Practicability) studies validate the practicability of a method, when it is applied by the target users instead of the method developers. Type III (effort–benefit) studies analyze the effort–benefit ratio of a prediction method by comparing the effort for conducting the same project at least twice: Once without using the prediction method, which may cause higher effort for rework, and once with using the prediction method, which may cause higher up-front effort.

For the sake of brevity, in this work we performed the Type I validation using BIMP tool[1] to simulate the business process generated according MABUP approach and compare with real life values. BIMP supports exponential, normal and uniform distributions for task execution times and for inter-creation times of process instances, which are the "representation of a single enactment" of a BP design. It also allows attaching calendars to resource pools in order to capture business hours. In addition generates statistics and histograms of execution times, costs and resource utilization that can be exported and analyzed them further in order to pinpoint bottlenecks.

4.2.1 Definition of Variables and Assumptions

We assumed that Organizational and Technological levels would have the same cost and time to be executed in both scenarios. Like this, we simulated only operational level. Concepts used in the modeling of the simulation involves:

- **Consumed Time**: It is the sum of consumed time in executing each task of the process since the beginning to its end considering all instances.
- **Cost**: It is the sum of cost of the execution of each task of the process since the beginning to its end considering all instances.
- **Workload:** The percentage value of the entire time the process has been executed, i.e. business process (BP) workload that specifies the intensity of the process execution by determining the amount of BP instances that traverse all the actor steps.

Beyond the operational modeling information indicated above, related to context variability and non-functional information, it is necessary to specify simulation information according to real life measurements obtained through interview. In this sense,

[1] available in *http://bimp.cs.ut.ee*

in **Table 9** we present the probability to occur of the variations expressed in **Figure 26**. We assume that the execution features of Organizational and Technological Levels are the same to both solutions.

Table 9 - Probability of occurrence (rounded)

Gateway	Decision	Probability
Data Lost Increasing	Froze board	5,00%
	Froze network gateway	5,00%
	Laser power degradation	90,00%
Restart Transmitter Fixed?	No	5,00%
	Yes	95,00%
Restart Gateway Fixed?	Yes	90,00%
	no <= 4 times	8,00%
	no > 4 times	2,00%

Additionally, a context variant event occurs approximately **every 10.7 hours**, ie. **804 occurrence per year**. The average value of human intervention to heal the system is R$ 55,00 per hour (for telecom engineer interventions in a centralized platform for monitoring/intervention) considering an Operating Expense (OPEX) rounded value to 10%. The tasks "Increase 1db the laser power of the transmitter" is executed manually in 30 minutes with an operational cost of R$ 27,50, and is executed automatically in less than 1 sec with a cost of R$ 0,55. The task "Restart transmitter board" is executed manually in 30 minutes with an operational cost of R$ 27,50, and is executed automatically in less than 1 sec with a cost of R$ 0,28. The task "Restart network gateway" is executed manually in 60 minutes with an operational cost of R$ 55,00, and is executed automatically in less than 10 minutes with a cost of R$ 5,50. The rest of the tasks have the fixed cost of R$ 0,80 and executed in less then 1 sec in both scenarios. **Table 10** presents the simulation modeling considering each task.

Table 10 - Simulation modeling of operational level of the task Allow Mobile Data Service

Variant	Task	Real World - Execution costs	Real World - Execution time	MABUP - Execution costs	MABUP - Execution time
Laser power gateway	Forward traffic to another interface	R$ 0,50	00:01h	R$ 1,00	00:01h
	Increase 1db the laser power of the transmitter	R$ 27,50	00:30h	R$ 0,55	00:00h
	Return traffic to main interface	R$ 0,50	00:02h	R$ 1,00	00:02h
Froze board	Forward traffic to another interface	R$ 0,50	00:01h	R$ 1,00	00:01h
	Restart transmitter board	R$ 27,50	00:30h	R$ 0,28	00:00h
	Return traffic to main interface	R$ 0,50	00:02h	R$ 1,00	00:02h
Froze network gateway	Forward traffic to another interface	R$ 0,50	00:01h	R$ 1,00	00:01h
	Restart network gateway	R$ 55,00	01:00h	R$ 5,50	00:10h
	Return traffic to main interface	R$ 0,50	00:02h	R$ 1,00	00:02h

Legend: RL - Real Life; M – MABUP

4.2.2 Simulation Design

In our experiment, we have investigated two hypothesis for comparing the solution alternatives, i.e., MABUP vs Real Life:

- Hypothesis 1: The concepts of MABUP, such as variation point, variant, context and operational task, covers simulation concepts such as cost, time and resources, which have to be supported by the domain modeling languages.
- Hypothesis 2: MABUP improve consumed time, cost and workload comparing with real life treatment.

4.2.3 Results

In **Table 11**, we present the results of one simulation execution. We can observe that *Consumed Time* of Manual Scenario is more than 1700% greater than Autonomic one because the engineer needs the time to think about all the information provided by the platform. In its turn, the *Total Cost* of execution in Manual Scenario is more than 1000% greater than the Autonomic one. Finally, the *workload* of Manual Scenario is more than 200% greater than Autonomic.

Table 11 - Cost, Time and Resource Consumption

Resource	Used scenario	Duration in days	Consumed Time	Total Cost	Workload
System	Autonomic Scenario	365d	1d 22:52h	R$4692,72	0,49%
Engineer	Manual Scenario	365d	20d 04:36h	R$49480,00	5,03%

Regarding hypothesis 1, we observed that the information modeled according MABUP approach reflected all operational information necessary to provide intervention in the system. The simulation reflects this observation even when we considered a sequence of replicated simulation runs. Using this approach, we easily correlated variation point as a specialization of complex gateways, context with an event and operational task as a specialization of service task. Furthermore, we could associate execution cost, execution time and resources to operational task; and highlighted how autonomic actions must be measured.

Regarding hypothesis 2, the simulation also confirms performance improvements of the solution applying MABUP approach compared to real life current practices.

4.3 Experimental Evaluation

As mentioned in Section 2.7, an experiment consists of a set of **tests** which each test is a combination of treatment, subject and object. These tests aims to verify if research hypotheses are true or false. Statistical analysis were used to reject null hypotheses, ie. when a hypothesis could not be confirmed. In order to evaluate concepts adopted in our MABUP approach, we have performed a controlled experiment that is detailed in sections below.

4.3.1 Experiment Design

- **Research Question:** Is the expressivity of autonomic features, modularity technique, structural complexity, behavioral similarity, modeling time and cognitive complexity of the process obtained though MABUP approach is the same of the one that do not use any approach?

Table 12 - Experiment Design

Experiment Design	
Goal	Analyze MABUP approach for the purpose of evaluation with respect to the representation of autonomic features in business process models.
General Objective	Evaluate the business process modeling of an organization that need to deal with autonomic features
Specific Objectives	We planed to evaluate the difference between the business process models with autonomic features produced using the MABUP approach and the one produced without the approach considering the following criteria: • Observation of expressivity of autonomic features • Observation of the use of separation of concerns • Measuring the use of modularity technique • Measuring structural complexity • Observation of behavioral similarity • Observation of cognitive complexity • Measuring the modeling time
Quantitative and Qualitative Indicators and Measurements	Autonomic Features • AF1 - Presence of non-functional requirements (NFR) and contexts *(Yes | No)* • AF2 - Presence of operational task to realize maintenance in the system if NFR deviations occurs *(Yes | No)* Separation of Concerns • SOC - Presence of multi levels in order to simplify the understandability of different concepts, ie. Business, operational and service information *(Yes | No)* Modularity Technique • NLM - Number of levels of the model Structural Complexity • NTM - Number of tasks in the model • NMM — Number of missing tasks or other elements in the model that represent the case study description • NSM - Number of subprocess in the model

Experiment Design	
	• NTL - Number of tasks in the same level with the majority number of elements • NEM - Number of events in the model • NGM - Number of gateways in the model <u>Behavioral Similarity</u> • BSC - Process behavior reflect the case study description *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)* • BS1 - Similarity with the model produced by the expert researcher in the organizational level - step 1 *(Yes | Similar, with max 1 task different | Similar, with max 2 tasks different | Similar, with more then 3 tasks different | No)* • BS2 - Similarity with the model produced by the expert researcher in the technological level - step 2 *(Yes | Similar, with max 1 task different | Similar, with max 2 tasks different | Similar, with more then 3 tasks different | No)* • BS3 - Similarity with the model produced by the expert researcher in the operational level - step 3 *(Yes | Similar, with max 1 task different | Similar, with max 2 tasks different | Similar, with more then 3 tasks different | No)* • BS4 - Similarity with the model produced by the expert researcher in the service level - step 4 *(Yes | No)* <u>Cognitive Complexity</u> • CC1 —Step 1 of MABUP modeling technique is comprehensive *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)* • CC2 — Step 2 of MABUP modeling technique is comprehensive *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)* • CC3 — Step 3 of MABUP modeling technique is comprehensive *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)* • CC4 — Step 4 of MABUP modeling technique is comprehensive *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)*

Experiment Design	
	<u>Modeling Time</u> • MDT — Modeling time *(hours and minutes - hh:mm)* <u>Easiness of use</u>
Experiment Context	In-vivo: we consider an experiment in the field under normal conditions with students of requirements engineering under graduate and graduate courses. Some subjects has industry experience. Subjects were given a problem description and tried the MABUP process to be solved it.
Sample Universe	34 (thirty four)
Number of evaluators	2 (two)
Experiment Session Duration	120 minutes

The design type adopted in this study is *one factor with two treatment, which* predicts the comparison of the means of dependable variables for each treatment. In this sense, we investigate if MABUP improves expressivity of autonomic features, modularization, complexity, behavioral similarity and modeling time than the previously modeling method. The factor is modeling method and the treatments are BPMN applying MABUP and BPMN without MABUP.

4.3.2 Hypothesis

As mentioned in the previous section, each metrics has two variations: using or not the MABUP approach. For example, there is modeling time using MABUP (MDTm) and the same metric not using MABUP (MDTc). The following hypotheses definition uses these symbols.

The main hypothesis is the null hypothesis that states there is no difference between using or not the MABUP approach. Therefore, the study tries to reject this hypothesis.

There are twenty null hypotheses, one for each metric the study analyzes as can be shown in **Table 13**. The following is a composition of these twenty null hypotheses.

Table 13 - Null and Alternative Hypothesis

Null Hypothesis	Alternative Hypothesis	Alternative Hypothesis
$H0_{1}$: AF1m $\cong$ AF1c	$H0_{1}$: AF1m > AF1c	$H0_{1}$: AF1m < AF1c
$H0_{2}$: AF2m $\cong$ AF2c	$H0_{2}$: AF2m > AF2c	$H0_{2}$: AF2m < AF2c
$H0_{3}$: SOCm $\cong$ SOCc	$H0_{3}$: SOCm > SOCc	$H0_{3}$: SOCm < SOCc
$H0_{4}$: NLMm $\cong$ NLMc	$H0_{4}$: NLMm > NLMc	$H0_{4}$: NLMm < NLMc
$H0_{5}$: NTMm $\cong$ NTMc	$H0_{5}$: NTMm > NTMc	$H0_{5}$: NTMm < NTMc
$H0_{6}$: NMMm $\cong$ NMMc	$H0_{6}$: NMMm > NMMc	$H0_{6}$: NMMm < NMMc
$H0_{7}$: NSMm $\cong$ NSMc	$H0_{7}$: NSMm > NSMc	$H0_{7}$: NSMm < NSMc
$H0_{8}$: NTLm $\cong$ NTLc	$H0_{8}$: NTLm > NTLc	$H0_{8}$: NTLm < NTLc
$H0_{9}$: NEMm $\cong$ NEMc	$H0_{9}$: NEMm > NEMc	$H0_{9}$: NEMm < NEMc
$H0_{10}$: NGMm $\cong$ NGMc	$H0_{10}$: NGMm > NGMc	$H0_{10}$: NGMm < NGMc
$H0_{11}$: BSCm $\cong$ BSCc	$H0_{11}$: BSCm > BSCc	$H0_{11}$: BSCm < BSCc
$H0_{12}$: BS1m $\cong$ 3	$H0_{12}$: BS1m > 3	$H0_{12}$: BS1m < 3
$H0_{13}$: BS2m $\cong$ 3	$H0_{13}$: BS2m > 3	$H0_{13}$: BS2m < 3
$H0_{14}$: BS3m $\cong$ 3	$H0_{14}$: BS3m > 3	$H0_{14}$: BS3m < 3
$H0_{15}$: BS4m = 0	$H0_{15}$: BS4m = 1	$H0_{15}$: BS4m = -1
$H0_{16}$: MDTm $\cong$ MDTc	$H0_{16}$: MDTm > MDTc	$H0_{16}$: MDTm $\cong$ MDTc
$H0_{17}$: CC1m $\cong$ 3	$H0_{17}$: CC1m > 3	$H0_{17}$: CC1m < 3
$H0_{18}$: CC2m $\cong$ 3	$H0_{18}$: CC2m > 3	$H0_{18}$: CC2m < 3
$H0_{19}$: CC3m $\cong$ 3	$H0_{19}$: CC3m > 3	$H0_{19}$: CC3m < 3
$H0_{20}$: CC4m $\cong$ 3	$H0_{20}$: CC4m > 3	$H0_{20}$: CC4m < 3

Independent Variables

Modeling Technique = {MABUP, Ad Hoc}

Dependent Variables = {AF1, AF2, SOC, NLM, NTM, NMM, NSM, NTL, NEM, NGM, BSC, BS1, BS2, BS3, BS4, MDT, CC1, CC2, CC3, CC4}

4.3.3 *Experiment Strategy*

The participants of this study are undergraduate, MSc and PhD students taking a graduate course on requirements engineering. The participants were aware that they were participating of an experimental study and that their data would be used in the study analysis.

Before starting the study, the subjects answered the questionnaire (**Appendix C**) about their profiles and experience with requirements engineering and software development in order to reduce the impact of experience factor in the experiment and provide a balanced design. The subjects were aware that their data would be used by the experimental study. The subject's data is presented in Section 4.3.4.

In order to identify the impacts of using the proposed method, the control object is the generation of the experimental object without the proposed method. In this way, the subjects are divided in two groups: the subjects that model business process using the proposed method (MABUP approach), and the subjects that do not use the proposed method.

As previously mentioned, this study was performed during a graduate course and consisted of two phases. In the first phase, we taught the business process modeling, non-functional requirements and contextualization. We verified if they learned above topics though project in which they had to create a business process model considering a requirements specification we wrote. Besides the project modeling, we performed oral argumentation with each subject about the project and business process modeling.

The second phase consisted in training the students about the proposed method and performing a dry run to give the subjects a chance to familiarize with steps of MABUP approach. Finally, the experiment was executed. Each activity spent the time listed in **Table 14**.

Table 14 - Time spent in each activity.

Activity	Time spent
Classes about: Business process modeling, non-functional requirements and contextualization	6hrs
Project and Oral argumentation	2hrs
Training about the process	2hrs
Exercises and projects evaluation	10h
Dry run	2hrs
Experiment	2hrs
Data Analysis	20h
Total	**44hrs**

We provided the business specification for both groups. Besides, the group with the treatment also received a document with the steps of the proposed method. The study object was the business process modeling of *'Telecom Service Management'* (**Appendix D and E**). They used BPMS Activiti with an extension with MABUP concepts (used just in MABUP group), described in **Appendix B.** After the experiment, subjects answered a post-experiment questionnaire **(Appendix F and G)** that was further analyzed.

The study analysis compares the collected data from the experimental object trials and control object trials in order to see if the null hypotheses can be rejected. The tests used to refute null hypothesis were used as presented in **Table 15**.

Table 15 - Statistical tests used in the experiment

<table>
<tr><th>Objective</th><th>Scale</th><th>Normality Test</th><th>Data constraint</th><th>Statistical comparison</th><th>Variance Test</th><th>Comparative Test</th><th>Evaluated Variables</th></tr>
<tr><td rowspan="4">Compare two independent sample</td><td rowspan="3">Metric</td><td rowspan="3">Anderson-Darling</td><td rowspan="2">Normal</td><td rowspan="2">Mean</td><td rowspan="2">F-test</td><td>T-test (unequal variance)</td><td rowspan="3">NLM, NTM, NMM, NSM, NTL, NEM, NGM, MDT</td></tr>
<tr><td>T-test (equal variance)</td></tr>
<tr><td>Non-normal</td><td>Median</td><td>-</td><td>U-test (Mann-Whitney or Wilcoxon rank sum test)</td></tr>
<tr><td>Non-metric (binominal)</td><td>-</td><td>Non-normal</td><td>Proportional</td><td>-</td><td>Chi-square goodness of fit</td><td>AF1, AF2, SOC</td></tr>
<tr><td rowspan="2">Compare one sample with a specific value</td><td rowspan="2">Metric or Non-metric</td><td rowspan="2">-</td><td rowspan="2">Non-normal</td><td>Median</td><td>-</td><td>Wilcoxon signed-rank test</td><td>CC1, CC2, CC3, CC4, BSC, BS1, BS2, BS3</td></tr>
<tr><td>Proportional</td><td>-</td><td>Chi-square Pearson test</td><td>BS4</td></tr>
</table>

4.3.4 Subjects

In the beginning of the experiment, the subjects answered the questionnaire (**Appendix C**) about their profiles and experience with requirements engineering and software development in order to reduce the impact of experience. In this sense, they were balanced according academic education, after were randomized divided. **Figure 27** shows the distribution of the groups according academic education.

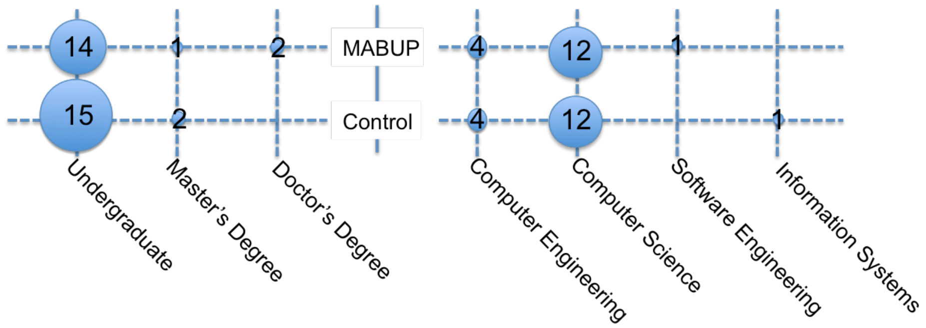


Figure 27 - Subject's degrees and academic education

Regarding with business modeling expertise, majority subjects answered that they had experience with BPMN and UML but they were not proficient in their adoption, as summarized in **Figure 28**.

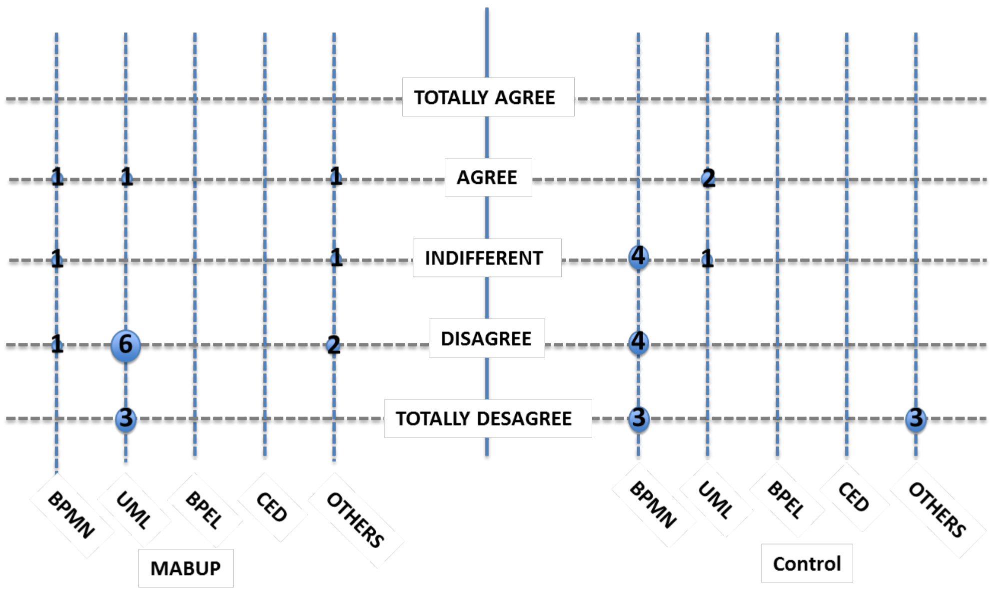


Figure 28 - Business Modeling Notations Experience and Proficiency

Considering another way to represent business description, in MABUP group 18% agree that they have proficiency on use that technique, 41% nor agree nor disagree and 41% disagree in their proficiency. Regarding Control group, there is a similar behavior with 23% that agree in their proficiency, 36% nor agree nor disagree and 41% disagree. **Figure**

29 presents the distribution per group of the proficiency in another way to represent business description.

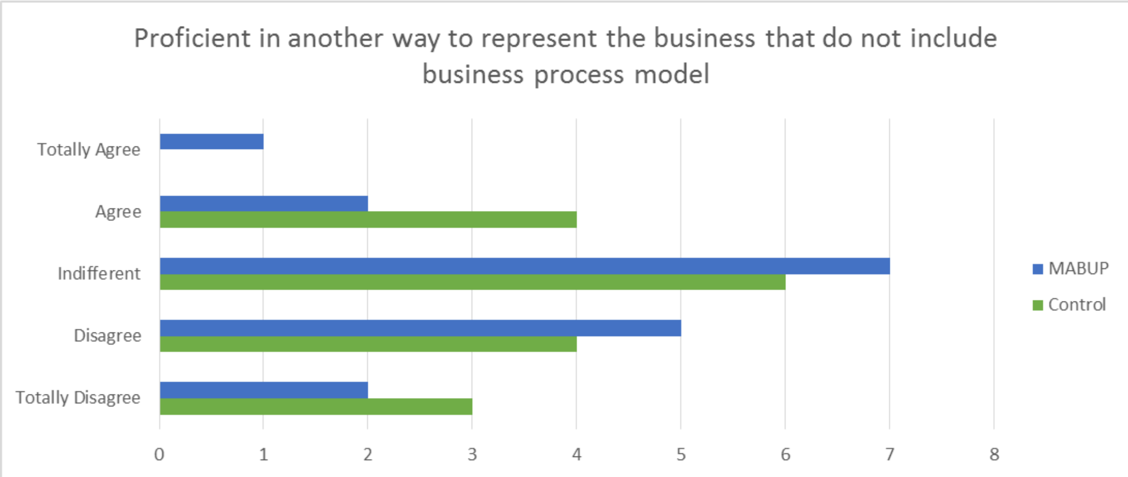


Figure 29 - Another Business Analysis Experience

Considering proficiency in system analysis and design, both groups were well balanced and have the same behavior with 29% of the subjects that agree in their proficiency, 18% that nor agree nor disagree and 53% that disagree as can be shown in **Figure 30**.

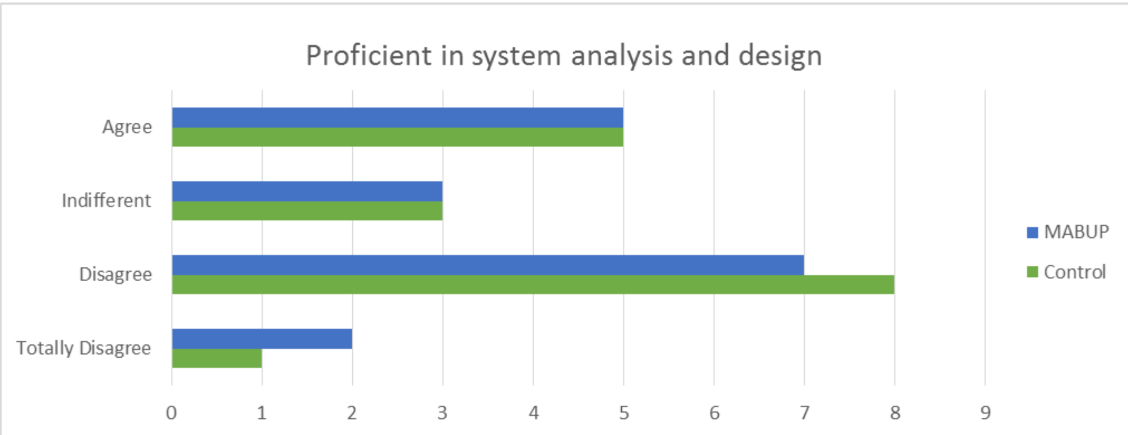


Figure 30 - Proficiency in system analysis

After even and coach the subjects with BPMN, NFR and contextualization knowledge; balance and randomize in groups; we asked all subjects to prepare and orally present a business modeling project with all these concepts based on a business description to evaluate if they were ready to realize the experiment. We determined a grade threshold to admit a subject in next steps of the experiment, additionally, absence and problems (sending corrupted files) in the execution of the experiment also discarded subject. In the end, we had 10 subjects in MABUP group and 13 subjects in Control group.

4.3.5 Results

In this section, we present and analyze the data collected through the observation of models produced by subjects or the questionnaires filled during the experiment for each group. In all case, we want to refuse null hypotheses with statistical tests as defined in **Table 15**. The significance used was 5%.

Autonomic Features

To check if the model provided autonomic features representation, we measured the: (i) presence of NFR and Context (AF1) and (ii) presence of operational tasks to realize maintenance in system if NFR deviation occurs (AF2). **Table 16** presents the results collected to each variable and **Figure 31** express percentage values.

Table 16 - Autonomic Features Results

Autonomic Features						
Group	**Presence of NFR and Context**			**Presence of Operational Tasks to realize maintenance in system if NFR deviation occurs**		
	Yes	**No**	**Total**	**Yes**	**No**	**Total**
MABUP	9	1	10	8	2	10
Control	6	7	13	9	4	13
Total	**15**	**8**	**23**	**17**	**6**	**23**

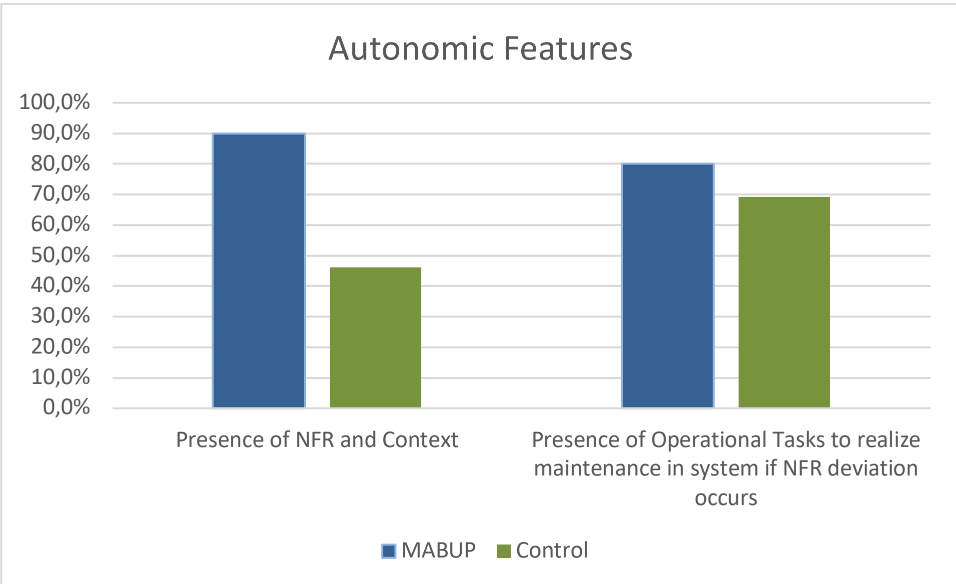


Figure 31 - Representation of Autonomic Features in Business Process Model

Considering these information, we used chi-square test to refuse null hypothesis, ie.: check if $H0_1$: AF1m ≘ AF1c and H02 : AF2m ≘ AF2c:

- AF1 - Presence of non-functional requirements (NFR) and contexts *(Yes | No)*

Table 17 presents the results of chi-square test to AF1 variable. Analyzing the p-value, we can affirm statistically that there is difference between modeling mode and the presence of NFR and context considering significance of 5%, ie. AF1 is a dependent variable and $H0_1$: AF1m > AF1c showing that MABUP group clearly expressed NFR and Context.

Table 17 - Chi-square test to AF1

Information	Value
X2 (AF1)=	4,790
gL	1
p-value	Value between 0,05 and 0,025
Significance	0,05

- AF2 - Presence of operational task to realize maintenance in the system if NFR deviations occurs *(Yes | No)*

Table 18 presents the results of chi-square test to AF2 variable. Analyzing the p-value, we can affirm statistically that there is no statistical difference between modeling mode and the presence of operational task to realize maintenance in the system if NFR deviations occur considering significance of 5%, ie. AF2 is not a dependent variable and $H0_2$: AF2m ≡ AF2c showing that both MABUP and Control group clearly expressed operational tasks to realize maintenance when exists a business description.

Table 18 - Chi-square test to AF2

Information	Value
X2 (AF2)=	2,086E+03
gL	1
p-value	Value between 0,975 e 0,95
Significance	0,05

We observed that AF1 (presence of NFR and Context) MABUP had a better evaluation compared to over the control group. This result shows that BPMN does not support these concepts, for this reason, the subjects of control group that expressed them had to use annotation in the model to represent the information. Regarding AF2 (presence of operational task to realize maintenance in the system if NFR deviation occurs), we observed that somehow the majority of subjects expressed operational information once the business description had this information. Given that, in real conditions, operational information is not addressed in business specification, then MABUP helps to think this issue in advance.

Separation of Concerns

- SOC - Presence of multi levels in order to simplify the understandability of different concepts, i.e. business, operational and service information *(Yes | No)*

To verify if the model provided separation of concerns, we measured the presence of multi levels in order to simplify the understandability of different concepts (SOC)**. Table 19** presents the results collected to this variable and **Figure 32** express percentage values.

Table 19 - Separation of Concerns (SOC) Results

Separation of Concerns			
Group	***Presence of multi levels in order to simplify the understandability of different concepts, ie. Business, operational and service information***		
	Yes	**No**	**Total**
MABUP	10	0	10
Control	4	9	13
Total	**14**	**9**	**23**

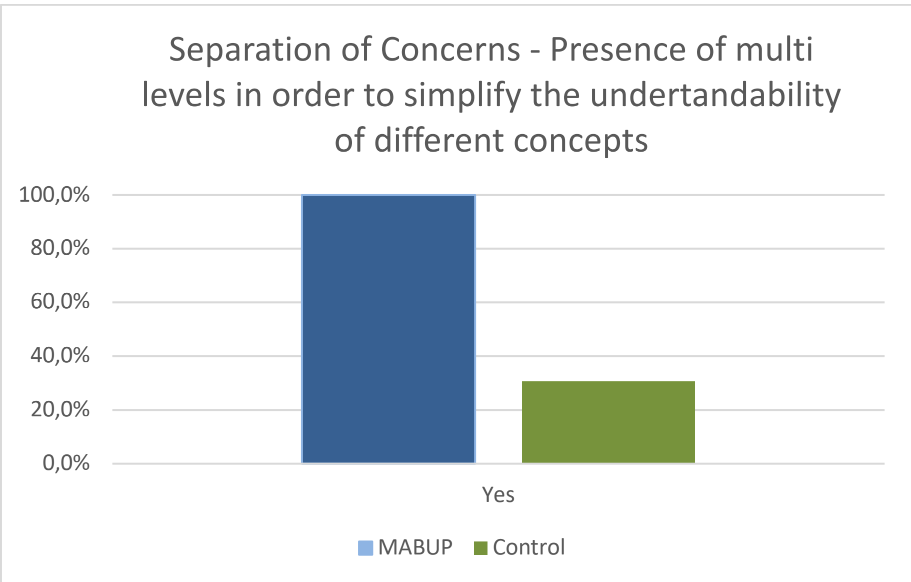


Figure 32 - Representation of Separation of Concerns in Business Process Model

Table 20 presents the results of chi-square test to SOC variable. Analyzing the p-value, we can affirm statistically that there is difference between modeling mode and the presence of multi levels in order to simplify the understandability of different concept considering significance of 5%, ie. SOC is a dependent variable and $H0_3$: SOCm > SOCc showing that MABUP group clearly expressed multi levels with different concepts.

Table 20 - Chi square test to SOC

Information	Value
X2 (SOC)=	11,374
gL	1
p-value	Value less than 0,005
Significance	0,05

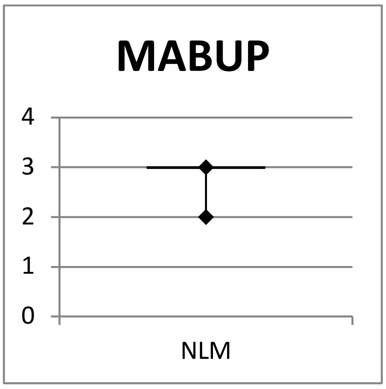
Modularity technique

- NLM - Number of levels of the model

To verify if the model provided modularity, we measured its number of levels. Initially, we realized a normality test (see **Table 15** and **Appendix I**) in order to verify data constraint. The test showed that the data were non-normal, so, the comparative test used was U-test. **Table 21** presents the basic statistic and test result to NLM values and **Figure 33** presents the distribution of this variable in both groups. Analyzing the p-value, we can conclude that there is a relationship between NLM and the modeling mode considering significance of 5%, ie. NLM is dependent and $H0_4$: NLMm > NLMc. We can observe that 9/10 subjects of MABUP group created 3 levels of the model as predicted in the modeling process, on the other hand, there was no pattern related control group. We also analyzed that the subject that modeled two levels using MABUP approach delivered an incomplete model.

Table 21 - Wilcoxon rank sum test for NLM Control and MABUP

Information	NLM-Control	NLM- MABUP
Maximum	3	3
Quartile 3	3	3
Mean	2,153846154	2,9
Median	3	3
Quartile 1	2	3
Minimum	1	2
Variance	0,474358974	0,1
Observations	13	10
W	104,5	
P-value	0,006109515	
Null Hypothesis (mu)	0	
(Pseudo) Median	0,999953595	
Inferior Limit	1,5643E-05	
Superior Limit	1,000026209	



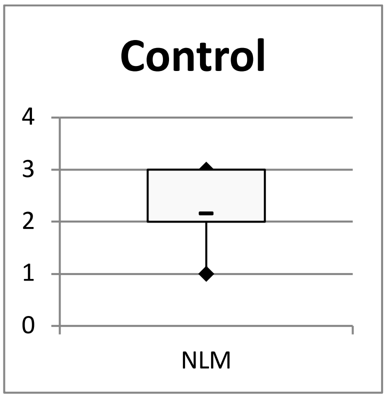


Figure 33 - Distribution of Number of Levels in the Model (NLM) in both groups

Structural Complexity

- NTM - Number of tasks in the model
- NMM — Number of missing tasks or other elements in the model that represent the case study description
- NSM - Number of subprocess in the model
- NTL - Number of tasks in the same level with the majority number of elements
- NEM - Number of events in the mode
- NGM - Number of gateways in the model

Initially, to verify structural complexity we realized a normality test (see **Table 15** and **Appendix I**) in order to verify data constraint. The test showed that there were non-normal data to NSM and NTL variables; so, the comparative test used was U-test. **Table 22** presents the basic statistic and test result to NSM and NTL values. Analyzing the p-value, we can conclude that there is no statistical difference between NSM and NTL ($H0_7$: NSMm ≅ NSMc and $H0_8$: NTLm ≅ NTLc) and the modeling mode although MABUP mean were lower.

Table 22 - Wilcoxon rank sum test for NSM and NTL Control and MABUP

Information	NSM - Control	NSM - MABUP	NTL - Control	NTL - MABUP
Maximum	2	**2**	29	**19**
Quartile 3	2	**1,75**	21	**16,75**
Mean	1,15	**1,3**	16,92	**15**
Median	1	**1**	15,5	**16**
Quartile 1	1	**1**	11	**14,25**
Minimum	0	**1**	8	**7**
Variance	0,4743589	**0,233333**	34,076923	**10,666666**
Observations	13	**10**	13	**10**
W	71,5		50,5	
P-value	0,666816986		0,382781518	
Null Hypothesis (mu)	-0,5975551		0	
(Pseudo) Median	4,18302E-05		-1,999940427	
Inferior Limit	-5,35301E-06		-5,999992851	
Superior Limit	0,999956559		3,000024542	

As mentioned NTM, NMM, NEM and NGM variables had normal metric, in this sense, we executed an F-test to verify if they have equal or unequal variance in order to execute the correct t-Test (see **Table 15** and **Appendix I**). We obtained that NTM and NMM had Unequal Variances and NEM and NGM Equal Variances. **Table 23** presents the basic statistic and test result of NTM, NMM, NEM and NGM variables; **Figure 34** and **Figure 35** presents the distribution of those variables in MABUP group and Control group respectively. Checking p-value, we conclude that NMM and NGM variables are dependent of modeling model. We can observe analyzing NMM and NGM variables that the mean and variance of control group is higher than MABUP group ($H0_6$: NMMm < NMMc and $H0_{10}$: NGMm < NGMc) which entails more incomplete and complexes models in control group over MABUP.

Table 23 - t-Test: Two-Sample Assuming Unequal Variances for NTM and NMM and Equal Variances for NEM and NGM Control and MABUP

Information	NTM - Control	NTM - MABUP	NMM - Control	NMM - MABUP
Maximum	39	**27**	11	**7**
Quartile 3	28	**26,75**	8	**5**
Mean	24,23	**23**	6,23	**4,2**
Median	21	**25**	3	**4**
Quartile 1	20	**21**	4	**3**
Minimum	15	**14**	3	**2**
Variance	41,0256410	**17,77777**	6,3589743	**3,2888888**
Observations	13	**10**	13	**10**
Hypothesized Mean Difference	0		0	
df	21		21	
t Stat	0,55410851		2,2452934	
P(T<=t) one-tail	0,29267875		0,0178217	
t Critical one-tail	1,720742903		1,7207429	
P(T<=t) two-tail	0,58535751		0,0356435	
t Critical two-tail	2,07961384		2,0796138	
Information	**NEM - Control**	**NEM - MABUP**	**NGM- Control**	**NGM - MABUP**
Maximum	19	**11**	14	**7**
Quartile 3	12	**9**	8	**5,75**
Mean	9,54	**8,1**	7	**4,4**
Median	8,55	**8**	4,2	**4**
Quartile 1	7	**7**	5	**3,25**
Minimum	4	**6**	3	**2**
Variance	22,6025641	**2,1**	11,166666	**2,933333**
Observations	13	**10**	13	**10**
Hypothesized Mean Difference	0		0	
df	15		19	
t Stat	1,03045814		2,4220835	
P(T<=t) one-tail	0,15956250		0,01279994	
t Critical one-tail	1,75305035		1,72913281	
P(T<=t) two-tail	0,31912501		0,0255998	
t Critical two-tail	2,13144954		2,0930240	

Regarding NTM and NEM variables, observing p-value we conclude that there are no statistical difference between those variables considering modeling mode, ie. they are not dependent ($H0_5$: NTMm ≅ NTMc and $H0_9$: NEMm ≅ NEMc). Even with this result, we can observe that the variables mean and variance of control group are higher then MABUP ones.

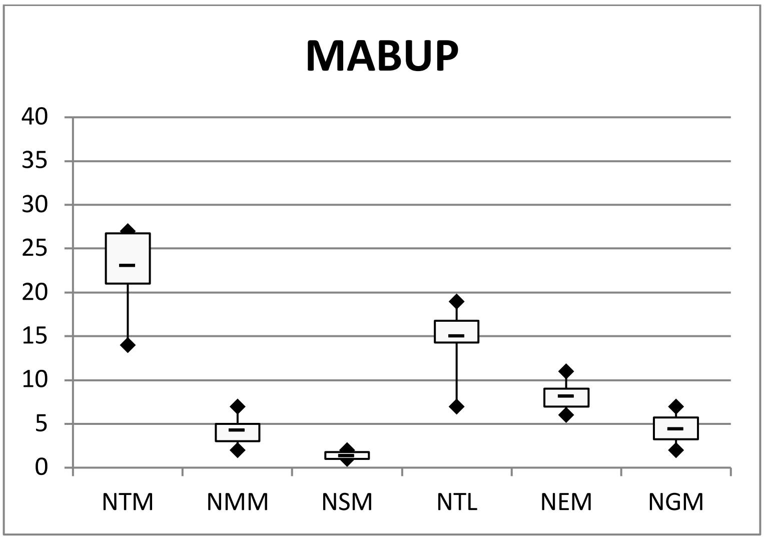


Figure 34 - Distribution of NTM, NMM, NSM, NTL, NEM and NGM in MABUP group

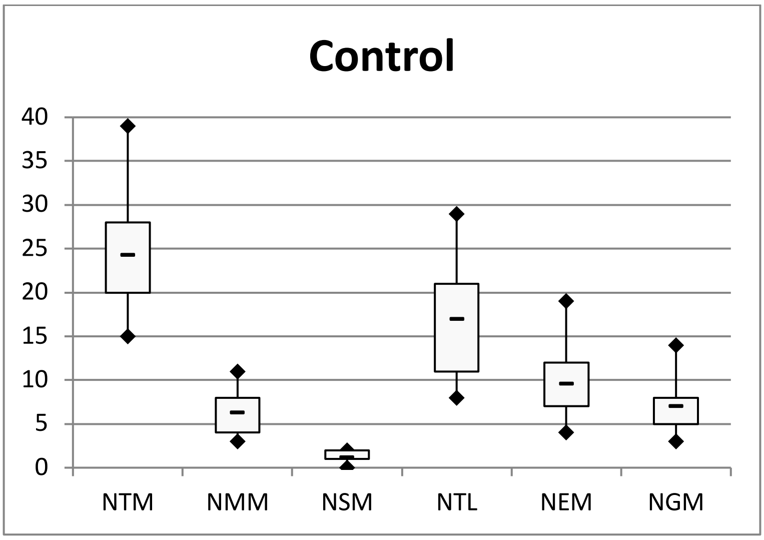


Figure 35 - Distribution of NTM, NMM, NSM, NTL, NEM and NGM in control group

Behavioral Similarity

The results of behavior similarity variables were collected through a comparison of the model generated by subjects and the model generated by researcher presented **Appendix H.**

- BSC - Process behavior reflect the case study description *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)*

The results obtained when analyzing variable BSC to both group are presented in **Table 24**. Observing these results we can conclude that both group had the same behavior to BSC variable, ie. ($H0_{11}$: BSCm ≅ BSCc showing that majority subjects were able to reflect business description in terms of business process models.

Table 24 - Behavior Similarity (BSC) Results

Behavior Similarity						
Process behavior reflect the case study description						
	Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree	Total
MABUP	5	4	0	0	1	10
Control	4	6	0	1	2	13

- BS1 - Similarity with the model produced by the research in the organizational level - step 1 *(Yes | Similar, with max 1 task different | Similar, with max 2 tasks different | Similar, with more then 3 tasks different | No)*
- BS2 - Similarity with the model produced by the research in the technological level - step 2 *(Yes | Similar, with max 1 task different | Similar, with max 2 tasks different | Similar, with more then 3 tasks different | No)*
- BS3 - Similarity with the model produced by the research in the operational level - step 3 *(Yes | Similar, with max 1 task different | Similar, with max 2 tasks different | Similar, with more then 3 tasks different | No)*
- BS4 - Similarity with the model produced by the research in the service level - step 4 *(Yes | No)*

The results obtained when analyzing the variables BS1, BS2, BS3 and BS4 are presented in **Table 25** and represented graphically in **Figure 36**. In order to statistically check BS1, BS2, BS3 results, we attributed ordinal scale as follow*: Yes = 5; Similar, with max 1 task different = 4; Similar, with max 2 tasks different = 3; Similar, with more than 3 tasks different =2; No = 1*; after, we executed U-test. Observing BS1 and BS3, we perceived that organizational and operational levels modeled by subjects were very similar to the expert research one showing that modularity technique was very effective and the identification of critical activities and operational tasks was quite simple. The variable BS2 shows that the modeling technique permits more freedom, which results in models similar but not equal to the expert research one. Almost all subjects indicated the services to autonomic tasks.

Table 25 - Behavior Similarity (BS1, BS2, BS3, BS4) Results

Behavior Similarity							
Process behavior reflect the case study description							
	Yes	**Similar, with max 1 task different**	**Similar, with max 2 tasks different**	**Similar, with more than 3 tasks different**	**No**	**Total**
BS1	2	4	4	0	0	10
BS2	0	2	2	4	2	10
BS3	8	0	1	0	1	10
BS4	8				2	10

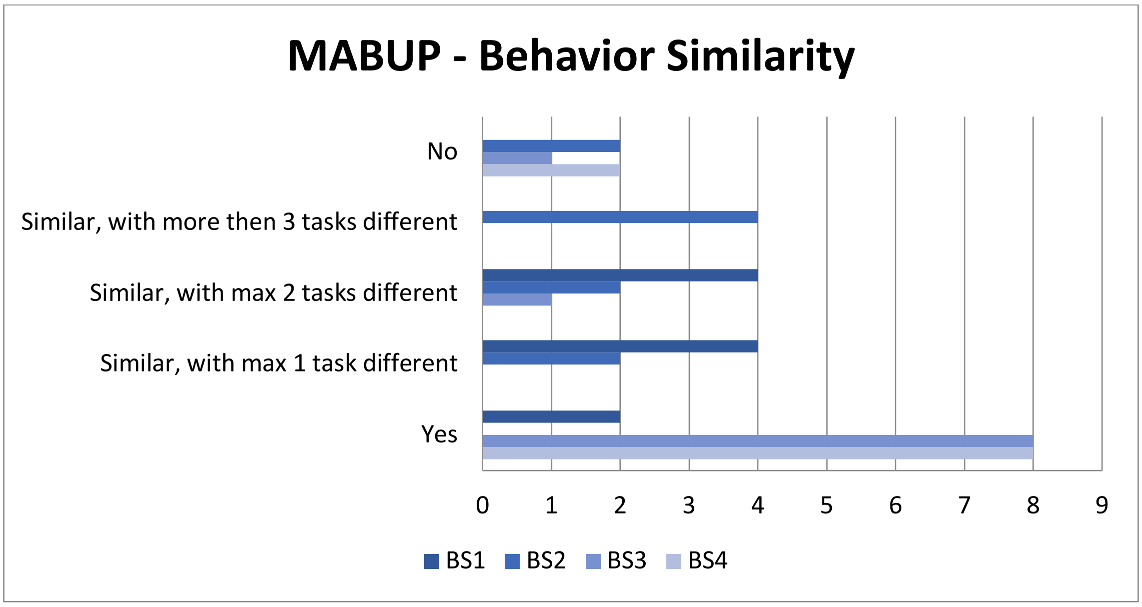


Figure 36 - Representation of Behavior Similarity on modeling MABUP

The results of U-test are presented in **Table 26**, considering the median 3 as null hypothesis to all variables and the results of Pearson test to BS4 in **Table 25**.

Analyzing p-value, we can refute null hypothesis to BS1 and BS3 variables ($H0_{12}$: BS1m >3 and $H0_{14}$: BS3m >3) and indicate that organizational and operational levels modeled by MABUP group were very similar to expert researcher. Considering the median, the null hypothesis cannot be refuted to variable BS2, so ($H0_{13}$: BS2m $\cong$ 3).

Regarding BS4, although did not present statistical difference ($H0_{15}$: BS4m = 0) we observed that subjects described service information in 80% of the sample.

Table 26 - Wilcoxon rank sum test for Cognitive complexity (BS1, BS2, BS3)

Information	*BS1*	*BS2*	*BS3*
V	21	7	40
P-value	0,023140931	0,106718816	0,019630657
Null Hypothesis (mu)	3	3	3
(Pseudo) Median	4,499998392	2,000034489	5
Inferior Limit	4	1,49996924	4,999920835
Superior Limit	4,500006	3,000029282	5

Table 27 - Pearson statistic to BS4

Information	*Value*
Pearson test	3,6
gL	1
p-value	0,057779571
Significance	0,05

Modeling Time

- MDT — Modeling time *(hours and minutes - hh:mm)*

In order to analyze modeling time, we measured the time that the subjects executed the modeling activity. Initially, we performed a normality test (see **Table 15** and **Appendix I**) in order to verify data constraint. The test showed that the data had normal distribution, so, the comparative test used was t-Test. **Table 28** presents the basic statistic and test result to MDT values and **Figure 37** presents the distribution of the variable in both groups. Analyzing the p-value, we can conclude that there is a relationship between MDT and the modeling mode considering significance of 5%, i.e. NLM is dependent.

Table 28 - t-Test: Two-Sample Assuming Unequal Variances for MDT Control and MABUP

Information	**MDT- Control**	**MDT- MABUP**
Maximum	2:15:00	2:21:00
Quartile 3	2:10:00	2:10:45
Mean	2:00:42	2:09:12
Median	2:07:00	2:10:00
Quartile 1	1:56:00	2:07:15
Minimum	1:41:00	1:53:00
Variance	6,02322E-05	2,41984E-05
Observations	13	10
Hypothesized Mean Difference	0	
df	20	
t Stat	-2,224640227	
P(T<=t) one-tail	0,018889494	
t Critical one-tail	1,724718243	
P(T<=t) two-tail	0,037778989	
t Critical two-tail	2,085963447	

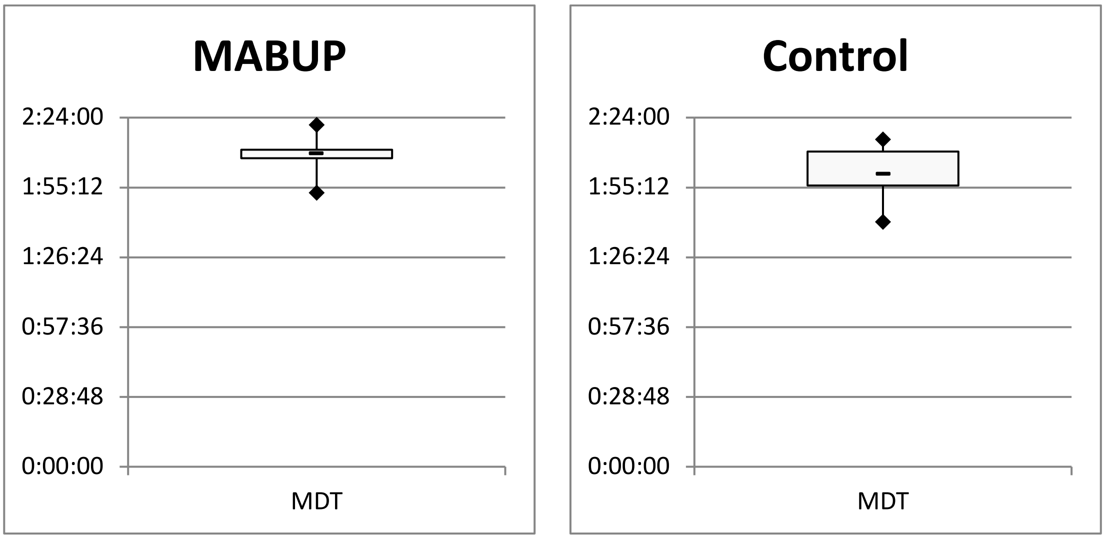


Figure 37 - Distribution of Modeling time (MDT) in both groups

Cognitive Complexity

- CC1 —Step 1 (Model organizational level) of MABUP modeling technique is comprehensive *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)*
- CC2 — Step 2 (Model technological level) of MABUP modeling technique is comprehensive *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)*
- CC3 — Step 3 (Model operational level) of MABUP modeling technique is comprehensive *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)*
- CC4 — Step 4 (Model service level) of MABUP modeling technique is comprehensive *(Strongly Agree | Agree | Neither agree nor disagree | Disagree | Strongly disagree)*

The results obtained from subjects post-experiment questionnaire regarding cognitive complexity when modeling with MABUP technique are presented in **Table 29** and represented graphically in **Figure 38**. In order to statistically check these results, we attributed ordinal scale as follow: *Strongly Agree = 5; Agree =4; Neither agree nor disagree = 3; Disagree = 2; Strongly disagree = 1*; after, we executed U-test. The results of the statistical test are presented in **Table 30**, considering the median 3 as null hypothesis to all variables. Analyzing p-value, we can refute null hypothesis to CC1, CC3 and CC4 variables and verify that organizational and operational levels (CC1 and CC3)) are comprehensive to be modeled once these levels have more defined rules and a minority knowledge to be represented. CC4 had a negative result once there were little information in the modeling description and training once it was not the focus of the experiment, i.e. the focus was business modeling. Regarding CC2, that did not present statistical difference, we observed that as technological level has the detail of business behavior, it is more complexes to be modeled.

Table 29 - Cognitive complexity (CC1, CC 2, CC 3, CC4) Results

Cognitive Complexity						
MABUP modeling technique is comprehensive						
	Strongly Agree	**Agree**	**Neither agree nor disagree**	**Disagree**	**Strongly disagree**	**Total**
CC1	7	2	0	1	0	10
CC2	2	3	1	4	0	10
CC3	5	3	1	0	1	10
CC4	0	0	5	3	2	10

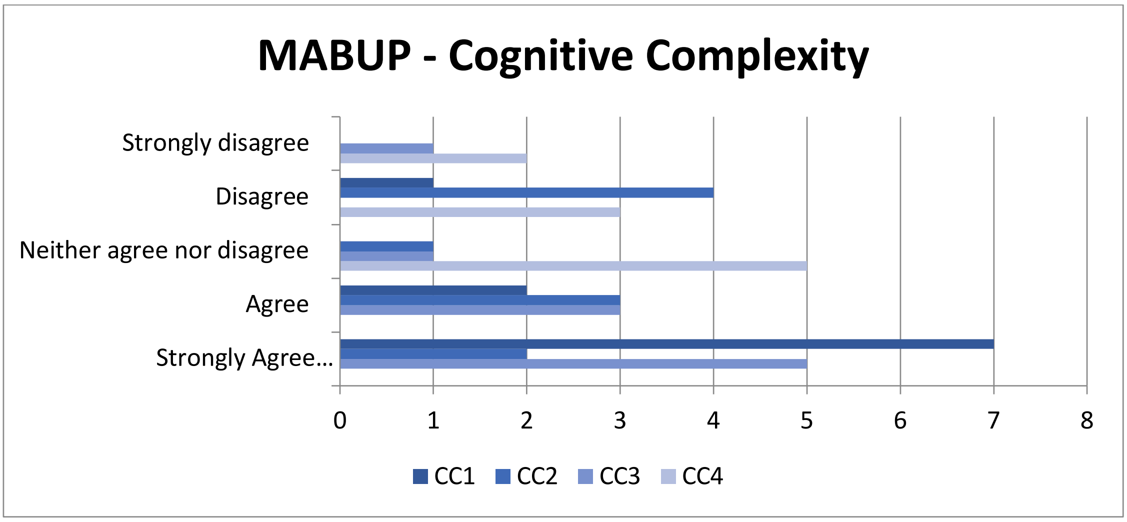


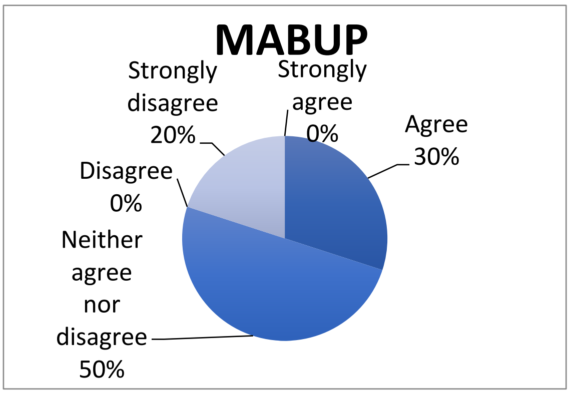
Figure 38 - Representation of Cognitive Complexity on modeling MABUP

Table 30 - Wilcoxon rank sum test for Cognitive complexity (CC1, CC 2, CC 3, CC4)

Information	CC1	CC2	CC3	CC4
V	53	29	38,5	0
P-value	0,006793539	0,416959373	0,049542558	0,03843393
Null Hypothesis (mu)	3	3	3	3
(Pseudo) Median	4,99993714	3,499997255	4,499999671	1,500042423
Inferior Limit	4,499979205	2	3,000021682	1,000064287
Superior Limit	5	4,499930377	5	2

Easiness of Use

In addition, analyzing subjects' post-experiment questionnaire regarding the easiness on model business description (with or without MABUP), **Figure 39** presents the results obtained.



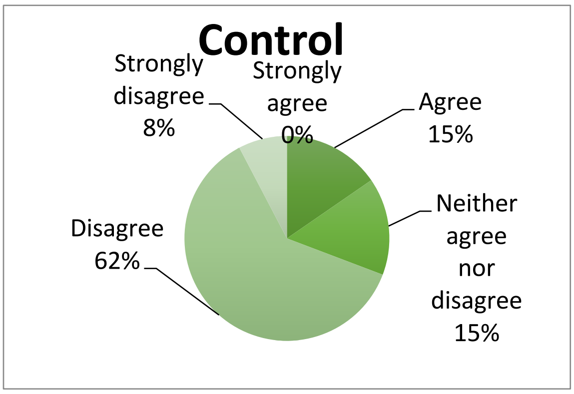


Figure 39 - Easiness of modeling business description in both groups

We observed that the two groups did not find easy to model in only 2 hours the problem considering the business description. However, the majority of MABUP group did not find difficulty to model the problem whereas majority of control group had difficulty.

4.4 Discussion

The experiment had the objective of analyze the MABUP approach from the point of view of autonomic features, modularity techniques, structural behavior, modeling time, cognitive complexity and easiness of use.

As far as **autonomic features** are concerned, we observed that in AF1 (presence of NFR and Context) MABUP had a better evaluation compared to over the control group. This result shows that BPMN does not support these concepts, for this reason, the subjects of control group that expressed them had to use annotation in the model to represent the information. Regarding AF2 (presence of operational task to realize maintenance in the system if NFR deviation occurs), we observed that somehow the majority of subjects expressed operational information once the business description had this information. Given that, in real conditions, operational information is not addressed in business specification, then MABUP helps to think this issue in advance.

Analyzing **separation of concerns** (variable SOC), we observed that majority subjects of control group used the same level to represent both business and operational knowledge. This feature showed that control group provided models harder to be understandable over MABUP group once there were different concerns treated in same levels. In the same way, verifying the use of **modularity** (variable NLM), we observed that 9/10 subjects of MABUP group modeled 3 levels of information with different propose, as predicted in the modeling process. On the other hand, the control group had no pattern (high variance) to modularize the information, showing that MABUP helps to modularize autonomic business process providing a level immune to technological changes.

Checking **structural complexity**, we observed that variables NTM, NSM, NTL and NEM did not have significant difference between MABUP and control group. This result shows that both group used similar number of task, subprocess and events to represent the business description. On the other hand, the control group had a higher number of missing elements (NMM) and gateways (NGM) in their description. These facts contributed to have more incomplete and complex model; and this fact can affect the other variables.

If we examine **behavior similarity** we note that, the variable BSC shows that most of the subjects had coherence in represent business description in both MABUP and control groups. Considering only MABUP group, that had a modeling process, we observed that organizational, operational and service level of the autonomic business process were very similar to the modeled by the expert researcher (results of variables BS1, BS3 and BS4) reinforcing that MABUP approach helps analysts to model autonomic features in business process models. Although we had expressive results for BS2, as the technological level has more information to be modeled, consequently, it is the largest level and thus the subjects has more freedom to represent these information in several ways, ie. can use different modeling elements (MABUP and core elements of BPMN) with different behavior to the same problem.

From the point of view of **cognitive complexity**, we observed that variables CC1, CC3 and CC4 had the null hypotheses refuted. Variables CC1 and CC3 had a positive result to MABUP group showing that the modeling technique is comprehensive to model organizational and operational levels. This result is consistent with the data obtained for previously analyzed variables and reinforces that these levels are more restricted in its construction and highlights elements that should be used. As the experiment did not focused in business process automation but just in discovering which tasks must be automated, this knowledge were addressed in a high level manner. We also observed that CC2 is not a dependent variable, this lead us to conclude that the technological level is modeled in a more free-form, particularly requiring the identification of tasks that need to be monitored for certain autonomic features.

Last but not least, examining the **modeling time**, we observed that MDT variable is a dependent variable and had a better result in control group. We identified that MABUP group spent more time to model once had to consult modeling technique and had to use a new notation that they had few familiarity. Furthermore, the modeling tool presented bug problems during the experiment.

The hypotheses results are presented in **Table 31** where we can observe where we could refute null hypotheses on the use of MABUP approach. Dependents variables are resumed in **Figure 40**, showing which of them had positive results for MABUP group and which were negative.

Table 31 - Hypotheses Results

Hypothesis Result	Comment
$H0_1$: AF1m > AF1c	Positive result to MABUP
$H0_2$: AF2m > AF2c	Positive result to MABUP
H03 : SOCm > SOCc	Positive result to MABUP
H04 : NLMm > NLMc	Positive result to MABUP
H05 : NTMm $\cong$ NTMc	Null hypothesis
H06 : NMMm < NMMc	Positive result to MABUP
H07 : NSMm $\cong$ NSMc	Null hypothesis
H08 : NTLm $\cong$ NTLc	Null hypothesis
H09 : NEMm $\cong$ NEMc	Null hypothesis
H010 : NGMm < NGMc	Positive result to MABUP
H011 : BSCm $\cong$ BSCc	Null hypothesis
H012 : BS1m >3	Positive result to MABUP
H013 : BS2m $\cong$ 3	Null hypothesis
H014 : BS3m >3	Positive result to MABUP
H015 : BS4m > 0	Null hypothesis
H016 : MDTm > MDTc	Negative result to MABUP
H017 : CC1m > 3	Positive result to MABUP
H018 : CC2m $\cong$ 3	Null hypothesis
H019 : CC3m > 3	Positive result to MABUP
H020 : CC4m < 3	Negative result to MABUP

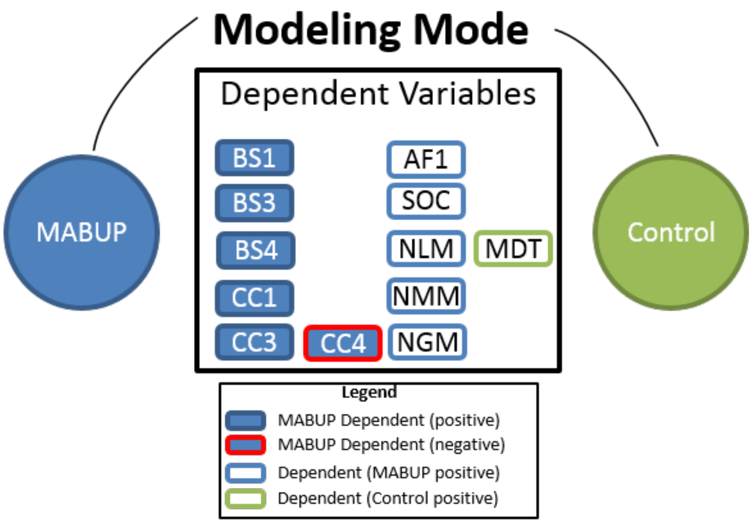


Figure 40 - Dependent Variables

Finally, analyzing the easiness on modeling the business description in 2 hours of experiments, we can observe that 70% of control group subjects disagree of the easiness against over 20% of MABUP. This result shows that it is feasible for the MABUP method to be adopted even with students and inexperienced analysts in business modeling.

4.4.1 Threats to Validity

This section discusses how valid are the results and if we can generalize them to a broad population. There are four kinds of validity. Internal validity defines if the collected data in the study resulted from the dependent variables and not from an uncontrolled factor. Conclusion validity is related to the ability to reach a correct conclusion about the collected data, to the used statistical test, and how reliable are the measures and the collected data. Construct validity is concerned to assure that the treatment reflects the cause and the results reflect the effect, for example, without being affected by human factors. Finally, external validity is concerned with the ability to generalize the results to an industrial environment.

Internal Validity

The experimental subjects are undergraduate, MSc and PhD students of an academic course. The students are from the software engineering area, so they had some experience in modeling software.

Despite being separated in two groups, one that used the proposed process and another that did not use, both subjects group received the same business description to be modeled. Therefore, we did not expect the subjects to be unhappy or discouraged in performing or not the treatment, since the resulting business process model should be equivalent behaviorally.

One confounding factor could be the subject's experience. In fact, the subjects filled a questionnaire about their experience and expertise in academia and industry. These data, presented in Section 4.3.4 were used in order to identify this possible confounding factor. Since there were few subjects to perform the study (34 subjects), we previously evaluated the experience and expertise, balanced considering this information and randomly distributed the subjects to treatments instead of defining blocks, which would decrease the number of samples to be compared.

Conclusion Validity

The experiment was based in a part of a real scenario and can represent problems that can affect autonomic systems. Anyway, it was only a part of the whole scenario that cannot be representative in terms of cognitive complexity. On the other hand, we have presented statistical tests to all variables to verify their dependence of modeling mode and analyzed the benefits of MABUP approach.

Construct Validity

The subjects applied the treatment to a specification of a '*Telecom Service Management'* process using an execution plan, which explains how to apply the treatment. In addition, they performed a dry run to make clear how the treatments should be implemented and how data should be collected.

External Validity

One expected result of this study is to guide analysts on when to use the proposed method. As we used balancing and randomization to separate the subjects in two groups, we expect to decrease the confounding factors, since the most important is the subjects' expertise.

Considering statistical tests used in this study, we can generalize some results with a significance of 5%. In this sense, we expect that the results, including the subjects' feedback, can be used as guidelines to improve the business process model. We also had problems with time assigned for the derivation, since the study was performed in one of the university laboratories, especially reserved for 2 hours for the execution of the experiment. We observed that some experiments were incomplete because of time restriction. Hence, as a result we could not obtain the real metrics to all cases analyzed.

Although the results are limited by the narrow scope we have, we believe that a considerable contribution is the study design. This experiment can guide other studies in order to evaluate the proposed method with more general and conclusive results and can also support other kind of studies, for example to identify the impact of other factors variation, evaluating alternative approaches to model business process.

4.5 CHAPTER SUMMARY

In this chapter, we presented an experiment that involved a simulation on the use of MABUP in a real scenario of a Brazilian telecommunication company. This study was important to highlight the benefits on the use of autonomic business-oriented solutions. We used a BPMN model to represent all MABUP concepts of MABUP approach and further model the simulation of operational level with real life measurements. The results were very positive, especially considering consumed time, cost and workload, which some values exceeded 1000% of improvement.

Moreover, an empirical evaluation with requirements engineering students showed that it is feasible for non-expert users to model guided with our proposal of multi-level autonomic business process approach, with promising results regarding expressivity of autonomic features, modularity technique, structural complexity, behavioral similarity and cognitive complexity when contrasted with an ad hoc approach. The next chapter presents final remarks about this work, including its limitations and future work.

5 Final Remarks

In this chapter we present a summary of our contributions and some final considerations, as well as future work to be carried on for the advancement of the MABUP approach.

5.1 CONTEXT

In order to deal with autonomic features, an interesting issue emerged to guide organization to survive with their complexity and business: autonomic business processes (ABP). The majority existing techniques to this solution addresses bind and re-bind of abstract activities to concrete services at run-time. In this sense, Lee et. al. (LEE et al., 2007a) discussed managing run-time adaptations in long running scientific workflows. In their work, adaptations have been described in terms of the functional decomposition of autonomic managers into monitoring, analysis, planning and execution components. Mosincat et al. (MOSINCAT; BINDER, 2008a) proposed fault tolerant execution of BPEL processes executing dynamic binding of services, performing a process transformation before the execution, for using a selection component at run-time. In Haupt et. al (HAUPT; SUKHIJA; ZHUK, 2011), the workflow and the services comprising it are treated as managed resources controlled by hierarchically organized autonomic managers. Their work attempts to treat the complexity problem of autonomic workflows by using hierarchical workflow but, it does not explore any modularity technique neither how these different levels can guide different kinds of adaptations. However, these works did not explore the modularization of business process that helps to treat the management of autonomic business process in complex scenarios and expressiveness of autonomic features in business process models;

In this thesis, we presented the MABUP - Multi Level Autonomic Business Process. It integrates business process models and MAPE architecture to provide autonomic adaptation to systems. Furthermore, it incorporates modeling techniques and new elements to business process modeling languages. In the following subsection the main contributions of this thesis are presented.

5.2 RELATED WORKS

Process that can be able to answer to AC principles are designated Autonomic Business Process (ABP) (RODRIGUES NT et al., 2007). Autonomic Business Process (or autonomic workflow) must have the capability to adjust to **environment variations** (context). If one **component service node** (self) becomes unavailable, a mechanism is needed to ensure a business process execution is not interrupted (YU; LIN, 2005). ABP differs from traditional workflow as it relies on autonomic techniques to manage adjustments during its execution. Therefore, it enables dynamic and automatic configuration of its definition, activities and resources. It also allows self-optimization and self-healing. Furthermore, autonomic workflow must have intelligence to analyze situations and deduce adaptations at run-time.

Strohmaaier and Yu in (STROHMAIER; YU, 2006) have presented the first attempt to apply autonomic computing principles to workflow management. Their work shows as certain levels of autonomy can already be achieved with available techniques and introduced the concepts of the different degree of "autonomy" in workflow systems but did not present a new technique.

In order to deal with autonomic features, an interesting issue emerged when composing applications: the ability to bind and re-bind abstract activities to concrete services at run- time. Several researchers have addressed this issue. Lee et. al. (LEE et al., 2007a) discussed managing run-time adaptations in long running scientific workflows. In their work, adaptations have been described in terms of the functional decomposition of autonomic managers into monitoring, analysis, planning and execution components. Mosincat et al. (MOSINCAT; BINDER, 2008a) proposed fault tolerant execution of BPEL processes executing dynamic binding of services, performing a process transformation before the execution, for using a selection component at run-time. In Haupt et. al. (HAUPT; SUKHIJA; ZHUK, 2011), the workflow and the services comprising it are treated as managed resources controlled by hierarchically organized autonomic managers. Their work attempts to treat the complexity problem of autonomic workflows by using hierarchical workflow but, it does not explore any modularity technique neither how these different levels can guide different kinds of adaptations.

The modularization helps to treat the complexity of large-size models (REIJERS, 2008). Once the management of business process models can be realized at different levels of granularity, there is a tradeoff between monitoring granularity and diagnostic precision. The finest level of monitoring granularity is at the functional level where all leaf level tasks are monitored. The disadvantage of fine-grained level monitoring is high monitoring and diagnostic overhead. Coarser levels of granularity only monitor higher-level business goals. In these cases, less complete and high-level monitoring data is generated, leading to multiple competing diagnoses. Both monitoring and diagnostic overheads are lower. The disadvantage is that if requirements denials are found, multiple diagnoses are returned, each pinpointing possible failures (WANG; MYLOPOULOS, 2009).

Generally, self-adaptive software is a closed-loop system with feedback from the self and the context. The Autonomic Computer System's building block, named autonomic manager, constantly interacts with managed element in order to maintain its equilibrium in face of incoming perturbations.

Monitoring the execution context represents another important aspect to implement an autonomic behavior and for reacting to events. However, even if, they exploit workflows to design applications and to specify causal constraints among activities and pages, the major works do not use contextual information in the autonomic business process management system (TRETOLA; ZIMEO, 2010).

Furthermore, these works do not explore another crucial issue that is the expressiveness of the operational knowledge in the business process. This problem is related to how the metrics impact in the management of business process and how to offer understandability of autonomic behavior.

The analysis of the related work shows that there is a lot of interest in the realization of systems and approaches able to deal, dynamically and automatically, with composition, binding, failures and other aspects of definition and execution of composed processes. However, majority of these works do not explore the modularization of business process that helps to treat the management of autonomic business process in complex scenarios and expressiveness of autonomic features in business process models. Furthermore, these works did not explore contextualization analyzes to guide proper adaptations.

Table 32 presents a comparative evaluation of autonomic business process approaches showing that MABUP covers problems found in actual works, i.e. Besides MABUP considers non-functional requirements, it deals with contextualization, modularization and the representation of autonomic features in business process models.

Table 32 - Comparative of autonomic business process approaches

	Self-awareness representation in BP	Context-awareness representation in BP	Modularization	Autonomic features represented in BP model	Autonomic features represented in autonomic manager	Autonomic features supported
(LEE et al., 2007b)	X	-	-	-	X	Self-optimization and self-healing
(GREENWOOD; RIMASSA, 2007)	X	-	-	X	-	Self-configuration and self-optimization
(MOSINCAT; BINDER, 2008b)	X	-	-	-	X	Self-healing
(TRETOLA; ZIMEO, 2010)	X	-	X	-	X	Self-configuration
(HAUPT; SUKHIJA; ZHUK, 2011)	X	-	X	-	X	Self-healing
MABUP	X	X	X	X	X	Self-configuration, self-healing, self-optimization and self-protection

5.3 CONTRIBUTIONS

In this proposal, we have presented a multi-level approach to model Autonomic Business Process, named MABUP. We argued that ABP must represent context-awareness and quality attributes to guide an appropriate adaptation. For this, we outlined:

I. A process to conduct the ABP modeling;

II. A meta-model that presented all concepts utilized in the MABUP approach;

III. An architecture of MABUP that explain the feasibility of our approach;

IV. An simulation to evaluate the benefits of the adoption of MABUP approach in a real scenario;

V. A controlled experiment to evaluate the benefits of the principles adopted in the design of MABUP approach.

Our approach consists of four separate but complementary levels of abstraction: Organizational Level, Technological Level, Operational Level and Service Level. These levels provide aspects to properly express important autonomic features such as variability by the use of contexts and operationalization of NFR.

Two real examples were used to illustrate the use of the approach. The presented aspects are supported by a MAPE (Monitor-Analyze-Plan-Execute) mechanism that considers both organizational context and the measurement of certain metrics. System metrics are critical for the monitoring, diagnosing deviations, planning and executing adaptations.

Considering that the expressiveness of autonomic computing features in business process model is still a poorly explored issue, the benefits of our approach are manifold and include addressing more expressivity, scalability and understandability in autonomic scenarios.

5.4 LIMITATIONS

Some may claim that the MABUP approach may be time consuming, as each element in the business process may experience several variations. Due to these traits, business analysts must use their tacit knowledge and the information about the business domain to explicit selection of proper adaptations. The process of eliciting autonomic business process has being explored for example in (TERRES; NT; SOUZA, 2010). The authors proposed a method to select business process for autonomic automation. Other possible solution is the use of trend analysis to identification of context expressions and the recurrent problems in their operational environment (VEDAM; VENKATASUBRAMANIAN, 1997).

In our running example, we only explored self-optimization principle of Autonomic Computing. Hence, we could not evaluate other principles to check if there was any specific impact on the modeling phase. Some approaches of self-configuration claims to adapt the business process model (eg. Santos et al. (2011)) but we did examine these kinds of approach.

5.5 FUTURE WORKS

We expect to continue this research project with the following improvements:

1. **Formalize description of concrete syntax of MABUP model to business process modeling tool**. In this work we provide a conceptual meta-model of MABUP concepts. In order to provide a tool with MABUP features it is necessary to formalize a concrete syntax according to the chosen business modeling language and its constraints. It is important to evaluate metrics about a complete adoption of MABUP approach on a real project.
2. **Other enhancements for the supporting tool**. We plan to investigate which improvements could be made to the MABUP extension in order to increase productivity and usability. Some of the features that could be developed are collaborative modeling, server-side saving and versioning, customized views, and tablet-tailored user interface.
3. **Further enhancement of the modeling technique.** We plan to address how the technological level modeling could be enhanced in order to provide more patterned models. These improvements are related with evidences obtained in the controlled experiment when we observed the difficulty of the subjects to model a business description with MABUP approach.
4. **Heuristics and guidelines.** In order to facilitate the use and improve the results obtained with this approach, further guidance can be provided on the decision-making aspects of the proposed modeling technique and system instrumentation.
5. **Adapt monitoring techniques to contemplate context information according business process.** Current monitoring techniques do not predict the obtainment of contextual information. It is necessary to provide techniques to guide the representation of contextual information in the management of operational environment.
6. **Further validation and improvements to handle large projects**. By applying MABUP approach to complete real solutions and projects, further limitations could be identified and new improvements could be devised.

5.6 Published Material

In Oliveira, Castro, Espana, & Pastor (2012), we present a process that exploits the high variability in service-oriented system environment by the use of contexts and autonomic adaptations by operationalization of Non-Functional Requirements (NFR).

In Oliveira, Castro, Santos, et al. (2012), we present the meta-model MABUPMM with the concepts used in our approach. The paper was nominee to best paper of the conference.

In Oliveira, Castro, España, & Pastor (2013), we introduce an architecture to support the autonomic management of business process using MAPE engine that explore MABUP model.

There are also other publications related to this work:

- OLIVEIRA, A. C.; DÓRA, P.; SPOHN, M.; OLIVEIRA, K. M. A.. From the Dark Net to the Cloudy Data: Cloud Network Governance Guidelines. XXXIV International Conference of the Chilean Society of Computer Science. Santiago, Chile: 2015 (OLIVEIRA et al., 2015): We conducted a survey to investigate how are the actual demands of customers, in order to provide useful information for the specification of new models and to support decision making in cloud businesses. After, we systematized an approach for cloud network governance to advance the state-of-the-practice in cloud governance aligned with business process management.
- FIGUEIREDO, B.; CASTRO, J.; OLIVEIRA, K. M. A. ”Uma Extensão de Elementos BPMN para Modelagem de Características Autonômicas em Processos de Negócio”. In: Workshop de Teses e Dissertações em Engenharia de Software (WTDSoft) held in CBSoft’2015 (FIGUEIREDO; CASTRO; OLIVEIRA, 2015): We presented a BPMS extended with MABUP concepts.
- OLIVEIRA, K. M. A., et al. "Representando Características Autonômicas nos Processos de Negócio." In Requirements Engineering @ Brazil, 2013, Rio de Janeiro, Brazil. CEUR Workshop Proceedings, 2013 (OLIVEIRA et al., 2013b): We introduces how to represent autonomic features in business process models;

- SANTOS, EMANUEL; PIMENTEL, J.; PEREIRA, T.; OLIVEIRA, K. M. A.; CASTRO, J. Business Process Configuration with NFRs and Context-Awareness. In Requirements Engineering @ Brazil (ER@BR 2013), 2013, Rio de Janeiro. CEUR Workshop Proceedings, 2013 (SANTOS et al., 2013): This paper presents a process to self-configure business process models considering NFR and context-awareness.
- BORATTO, MURILO ; ALONSO, PEDRO ; GIMÉNEZ, DOMINGO ; BARRETO, MARCOS ; OLIVEIRA, KAROLYNE . Auto-tuning methodology to represent landform attributes on multicore and multi-GPU systems. In: the 2013 International Workshop, 2013, Shenzhen. Proceedings of the 2013 International Workshop on Programming Models and Applications for Multicores and Manycores – PMAM'13. New York: ACM Press (BORATTO et al., 2013): This paper presents how to provide self-optimization on multicore and multi-GPU systems;
- OLIVEIRA, K. M. A. ; PIMENTEL, J. ; SANTOS, Emanuel ; DERMEVAL, D. ; GUEDES, G. ; SOUZA, C. ; SOARES, M. ; CASTRO, J. ; ALENCAR, F. ; SILVA, C. . 25 years of Requirements Engineering in Brazil: a systematic mapping. In: Conference on Software-Engineering in Iberoamerica, 2013, Montevideo. Workshop on Requirements Engineering (WER 2013), 2013. p. 118-134 (OLIVEIRA et al., 2013c): This paper presents a mapping review of the last 25 years of requirements engineering in Brazil .

References

ABETI, L. et al. Business Process Modeling for Organizational Knowledge Management. In: **Concurrency, Graphs and Models**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. v. 5065.

ABPMP. **Guide to the Business Process Management Body of Knowledge (BPM CBOK®)**. [s.l: s.n.].

BABAOGLU, O. **Self-Star Properties in Complex Information Systems: Conceptual and Practical FoundationsSelfstar properties in complex information systems conceptual and practical foundations Ozalp Babaoglu ed**Springer, , 2005.

BAINA, K.; BAINA, S. **User experience-based evaluation of open source workflow systems: The cases of Bonita, Activiti, jBPM, and Intalio**. 2013 3rd International Symposium ISKO-Maghreb. **Anais**...2013Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6728122>

BOHME, R.; REUNSSNER, T. Validation of predictions with measurements. 2008.

BORATTO, M. et al. **Auto-tuning methodology to represent landform attributes on multicore and multi-GPU systems**. Proceedings of the 2013 International Workshop on Programming Models and Applications for Multicores and Manycores - PMAM '13. **Anais**...New York, New York, USA: ACM Press, 2013Disponível em: <http://dl.acm.org/citation.cfm?doid=2442992.2443006>

BOX, D. et al. **Simple object access protocol (SOAP) 1.1**, 2000.

BUCKLEY, J.; MENS, T.; ZENGER, M. Towards a taxonomy of software change. **Journal of Software**, p. 1–7, 2005.

CAETANO, A.; SILVA, A. R.; TRIBOLET, J. An Application of the Separation of Concerns Principle to Business Process Design. **Int. J. Organisational Design and Engineering**, v. x, n. x, 2012.

CARDOSO, E.; ALMEIDA, J.; GUIZZARDI, G. **Eliciting goals for business process models with non-functional requirements catalogues**. CAiSE 2009. **Anais**...Amsterdam, The Netherlands: 2009Disponível em: <http://www.inf.ufes.br/~gguizzardi/BPMDS09.pdf>. Acesso em: 20 mar. 2012

CETINA, C.; PELECHANO, V. Variability in Autonomic Computing. In: CAPILLA, R.; BOSCH, J.; KANG, K.-C. (Eds.). . **Systems and Software Variability Management**. Berlin, Heidelberg, Heidelberg: Springer Berlin Heidelberg, 2013. p. 261–267.

CHANG, J. F. **Business Process Management Systems: Strategy and Implementation**. [s.l.] CRC Press, 2005.

CHRISTENSEN, E. et al. **Web services description language (WSDL) 1.1**, 2001.

COLIN BEVERIDGE. Aligning IT with business strategy. **Guidelines for IT Management**, n. 273, 2006.

CURBERA, F. et al. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. **Internet Computing, IEEE**, v. 6, n. 2, p. 86–93, 2002.

DOYLE, J.; FRANCIS, B.; TANNENBAUM, A. Feedback Control Theory. **Design**, v.

134, n. 6, p. 210, 1990.

ESPANA, S. et al. **Communication Analysis: A Requirements Engineering Method for Information Systems**. Proceedings of the 21st International Conference on Advanced Information Systems Engineering. **Anais**...: CAiSE ’09.Berlin, Heidelberg: Springer-Verlag, 2009Disponível em: <http://dx.doi.org/10.1007/978-3-642-02144-2_41>

EVESTI, A.; OVASKA, E. Comparison of Adaptive Information Security Approaches. **ISRN Artificial Intelligence**, v. 2013, p. 1–18, 2013.

FIGUEIREDO, B.; CASTRO, J.; OLIVEIRA, K. **Uma Extensão de Elementos BPMN para Modelagem de Características Autonômicas em Processos de Negócio**. Workshop de Teses e Dissertações em Engenharia de Software (WTDSoft) held in CBSoft’2015. **Anais**...2015

GAMMA, E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. Reading, MA: Addison-Wesley, 1995.

GANEK, A. G.; CORBI, T. A. The dawning of the autonomic computing era. **IBM Systems Journal**, v. 42, n. 1, p. 5–18, 2003.

GESTON, J.; NELIS, J. **Business process management**. [s.l.] Routledge, 2014.

GONZALEZ, A.; ESPANA, S.; PASTOR, O. **Unity criteria for Business Process Modelling**. Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on. **Anais**...2009

GREENWOOD, D.; RIMASSA, G. **Autonomic Goal-Oriented Business Process Management**. Third International Conference on Autonomic and Autonomous Systems (ICAS’07). **Anais**...Ieee, 2007Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4437920>

HAMMER, M. Reengineering work: don’t automate, obliterate. **Harvard business review**, v. 68, n. 4, p. 104–112, 1990.

HAUPT, T.; SUKHIJA, N.; ZHUK, I. **Autonomic execution of computational workflows**. Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on. **Anais**...2011

HINCHEY, M. G.; STERRITT, R. Self-Managing Software. **IEEE Computer**, v. 39, n. 2, p. 107–109, 2006.

HORN, P. **Autonomic Computing: IBM’s Perspective on the State of Information TechnologyIBM Corporation**: Tertiary Autonomic computing: IBM’s perspective on the state of information technology.IBM, , 2001. Disponível em: <http://www-1.ibm.com/industries/government/doc/content/resource/thought/278606109.html>

ISO/IEC 9126-1. **ISO/IEC 9126-1: Software engineering — Product quality — Part 1: Quality modelEnglish**. [s.l: s.n.].

JELASITY, M.; BABAOGLU, O. T-Man: Gossip-based overlay topology management. In: **Engineering Self-Organising Systems**. [s.l.] Springer, 2006. p. 1–15.

JESTON, J.; NELIS, J. **Business Process Management: Practical Guidelines to Successful Implementations**. [s.l: s.n.].

JORDAN, D.; ALVES, A. Web Services Business Process Execution Language Version 2 . 0. **Language**, v. 11, n. April, p. 1–264, 2007.

KELLY, S.; TOLVANEN, J.-P. **Domain-Specific Modeling: Enabling Full Code Generation**. [s.l.] Wiley, 2008.

KEPHART, J. O.; CHESS, D. M. The vision of autonomic computing. **Computer**, v. 36, n. 1, p. 41–50, jan. 2003.

KEPHART, J. O.; WALSH, W. E. **An artificial intelligence perspective on autonomic computing policies**. Proceedings. Fifth IEEE International Workshop on Policies for Distributed Systems and Networks, 2004. POLICY 2004. **Anais**...IEEE, 2004Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1309145>. Acesso em: 18 set. 2014

KEPHART, J.; THOMAS, I. B. M. **Autonomic Computing: The First Decade**. ICAC 2011 Keynote. **Anais**...2011

KOKAR, M. M.; BACLAWSKI, K.; ERACAR, Y. A. Control Theory-Based Foundations of Self- Controlling Software. **IEEE Intelligent Systems**, v. 14, n. 3, p. 37–45, 1999.

LADDAGA, R. Active Software. **SelfAdaptive Software**, Lecture Notes in Computer Science. v. 1936, p. 11–26, 2000.

LEE, K. et al. **Workflow adaptation as an autonomic computing problem**. Proceedings of the 2nd workshop on Workflows in support of large-scale science. **Anais**...: WORKS ’07.New York, NY, USA: ACM, 2007aDisponível em: <http://doi.acm.org/10.1145/1273360.1273366>

LEE, K. et al. **Workflow adaptation as an autonomic computing problem**. Proceedings of the 2nd workshop on Workflows in support of large-scale science. **Anais**...New York, NY, USA: ACM, 2007b

LUIS, J. et al. Business Processes Contextualisation via Context Analysis. **Context**, p. 1–6, 2010.

MAGGI, F. M. et al. Predictive Monitoring of Business Processes. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [s.l: s.n.]. v. 8484 LNCSp. 457–472.

MAIRIZA, D.; ZOWGHI, D.; NURMULIANI, N. An Investigation into the Notion of Non-Functional Requirements. p. 311–317, 2010.

MORENO, N.; ROMERO, J. R.; VALLECILLO, A. An Overview Of Model-Driven Web Engineering and the Mda. In: ROSSI, G. et al. (Eds.). . **Web Engineering**. Human-Computer Interaction Series. [s.l.] Springer, 2008. p. 353–382.

MOSINCAT, A.; BINDER, W. **Transparent Runtime Adaptability for BPEL Processes**. Proceedings of the 6th International Conference on Service-Oriented Computing. **Anais**...: ICSOC ’08.Berlin, Heidelberg: Springer-Verlag, 2008aDisponível em: <http://dx.doi.org/10.1007/978-3-540-89652-4_20>

MOSINCAT, A.; BINDER, W. **Transparent Runtime Adaptability for BPEL Processes**. Proceedings of the 6th International Conference on Service-Oriented Computing. **Anais**...Berlin, Heidelberg: Springer-Verlag, 2008b

OLIVEIRA, A. C. et al. **From the Dark Net to the Cloudy Data: Cloud Network Governance Guidelines**. XXXIV International Conference of the Chilean Society of Computer Science. **Anais**...Santiago, Chile: 2015

OLIVEIRA, K. et al. **Towards Autonomic Business Process Models**. International Conference on Software Engineering and Knowledge (SEKE 2012). **Anais**...San Francisco, California, USA: Knowledge Systems Institute Graduate School, 2012aDisponível em: <http://dblp.uni-trier.de/db/conf/seke/seke2012.html#OliveiraCEP12>

OLIVEIRA, K. et al. **A Multi Level Approach to Autonomic Business Process**. 2012 26th Brazilian Symposium on Software Engineering. **Anais**...Natal, RN: IEEE, set. 2012bDisponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6337862>. Acesso em: 15 ago. 2013

OLIVEIRA, K. et al. Multi-level Autonomic Business Process Management. **Enterprise, Business-Process …**, n. i, p. 184–198, 2013a.

OLIVEIRA, K. et al. **Representando Características Autonômicas nos Processos de Negócio**. Requirements Engineering @ Brazil. **Anais**...CEUR Workshop Proceedings, 2013bDisponível em: <http://ceur-ws.org/Vol-1005/erbr2013_submission_38.pdf>

OLIVEIRA, K. et al. **25 years of Requirements Engineering in Brazil: a systematic mapping**. Workshop on Requirements Engineering. **Anais**...2013c

OMG. **OMG Unified Modeling Language (OMG UML), InfrastructureInformatikSpektrum**. [s.l: s.n.]. Disponível em: <http://www.omg.org/cgi-bin/doc?formal/09-02-04>.

OMG. Business Process Model and Notation (BPMN). **Business**, v. 1, n. August 2009, p. 63–316, 2010b.

OREIZY, P.; MEDVIDOVIC, N.; TAYLOR, R. N. **Architecture-based runtime software evolution**. Proceedings of the 20th International Conference on Software Engineering. **Anais**...: ICSE ’98.IEEE Comput. Soc, 1998Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=671114>

PAPAZOGLOU, M. P. et al. Service-oriented computing: State of the art and research challenges. **Computer**, n. 11, p. 38–45, 2007.

PARASHAR, M.; HARIRI, S. Autonomic computing: An overview. **Unconventional Programming Paradigms**, p. 247–259, 2005.

PAVLOVSKI, C. J.; ZOU, J. Non-Functional Requirements in Business Process Modeling. **Reproduction**, APCCM ’08. v. 79, n. Apccm, p. 103–112, 2008.

RADEMAKERS, T.; BARREZ, J. **Activiti in Action**. [s.l: s.n.].

RAHMAN, M. et al. A taxonomy and survey on autonomic management of applications in grid computing environments. **Concurr. Comput. : Pract. Exper.**, v. 23, n. 16, p. 1990–2019, nov. 2011.

REIJERS, H. Modularity in process models: Review and effects. **Business Process Management**, p. 20–35, 2008.

RODRIGUES NT, J. A. et al. **Autonomic Business Processes Scalable Architecture**. Proceedings of the BPM 2007 International Workshops BPI BPD CBP ProHealth RefMod semantics4ws. **Anais**...Springer - Verlag Berlin Heidelberg, 2007

RUIZ, M.; ESPAÑA, S.; GONZÁLEZ, A. **Análisis de Comunicaciones como un enfoque de requisitos para el desarrollo dirigido por modelos**. DSDM 2010. **Anais**...2010

SALEHIE, M. et al. **Change Support in Adaptive Software: A Case Study for Fine-Grained Adaptation**. 2009 Sixth IEEE Conference and Workshops on Engineering of Autonomic and Autonomous Systems. **Anais**...Ieee, 2009Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4839203>

SALEHIE, M.; TAHVILDARI, L. Autonomic computing: emerging trends and open problems. **Architecture**, v. 30, n. 4, p. 1–7, 2005.

SALEHIE, M.; TAHVILDARI, L. **Self-Adaptive software: Landsacape and research challenges**. ACM Transactions on Autonomous and Adaptive Systems TAAS. **Anais**...ACM, 2009Disponível em: <http://portal.acm.org/citation.cfm?id=1516538>

SANTOS, E. et al. Using NFR and context to deal with adaptability in business process models. **2011 2nd International Workshop on Requirements@Run.Time**, p. 43–50, ago. 2011.

SANTOS, E. et al. **Business Process Configuration with NFRs and Context-Awareness**. Requirements Engineering @ Brazil (ER@BR 2013). **Anais**...CEUR Workshop Proceedings, 2013

SENDEROVICH, A. et al. **Queue mining - Predicting delays in service processes**. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). **Anais**...2014

SMANCHAT, S.; LING, S.; INDRAWAN, M. **A survey on context-aware workflow adaptations**. Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia. **Anais**...: MoMM ’08.New York, NY, USA: ACM, 2008Disponível em: <http://doi.acm.org/10.1145/1497185.1497274>

STROHMAIER, M.; YU, E. **Towards autonomic workflow management systems**. [s.l.] ACM Press, 2006.

TELES, F.; LINS, F.; ROSA, N. Towards a model-driven approach for treating quality attributes of service-based business processes. **International Journal of Web Engineering and Technology**, v. 9, n. 4, p. 330, 2014.

TERRES, L. D.; NT, J. A. R.; SOUZA, J. M. DE. Selection of Business Process for Autonomic Automation. **2010 14th IEEE International Enterprise Distributed Object Computing Conference**, p. 237–246, out. 2010.

TORRES, R. C.; MARTINS, L. E. G. NFR Catalogues for RFID Middleware. v. 14, n. 2, p. 102–108, 2014.

TRETOLA, G.; ZIMEO, E. **Autonomic internet-scale workflows**. Proceedings of the 3rd International Workshop on Monitoring, Adaptation and Beyond. **Anais**...: MONA ’10.New York, NY, USA: ACM, 2010Disponível em: <http://doi.acm.org/10.1145/1929566.1929573>

VAN DER AALST, W. M. P.; TER HOFSTEDE, A. H. M.; WESKE, M. Business process management: A survey. In: **Business process management**. [s.l.] Springer, 2003. p. 1–12.

VEDAM, H.; VENKATASUBRAMANIAN, V. **A wavelet theory-based adaptive trend analysis system for process monitoring and diagnosis**. American Control Conference, 1997. Proceedings of the 1997. **Anais**...jun. 1997

WANG, Y.; MYLOPOULOS, J. Self-Repair through Reconfiguration: A Requirements Engineering Approach. **2009 IEEE/ACM International Conference on Automated Software Engineering**, p. 257–268, nov. 2009.

WEBER, H. **Modularity in Data Base System Design: A Software Engineering View of Data Base Systems.** VLDB Surveys. **Anais**...1978

WHITE, S. A.; MIERS, D. **BPMN Modeling and Reference Guide**. [s.l.] Lighthouse Point, Florida, 2008.

WIERINGA, R.; MORALI, A. Technical Action Research as a Validation Method in Information Systems Design Science. In: **Design Science Research in Information Systems. Advances in Theory and Practices**. [s.l: s.n.]. v. 7286p. 52–65.

WOHLIN et al. **Experimentation in software engineering: an introduction**. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

YAU, S.; YAN, M.; HUANG, D. Design of Service–Based Systems with Adaptive Tradeoff Between Security and Service Delay. **Autonomic and Trusted Computing**, v. NA, p. 103–113, 2007.

YU, T.; LIN, K. **Adaptive algorithms for finding replacement services in autonomic distributed business processes**. Proceedings Autonomous Decentralized Systems 2005 ISADS 2005. **Anais**...: Proceedings - 2005 International Symposium on Autonomous Decentralized Systems, {ISADS} 2005.Ieee, 2005Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1452105>

Appendix A - Using MABUP with Communication Analysis

Modeling Phase - Extending Communicative Event Diagram (CED) and Business Process Modeling Notation (BPMN)

In order to examine the application of MABUP concepts in a business process modeling language, we examined Communicative Event Diagram (CED) to model a simplified version of CAGED (See Section 3.3). We analyzed that CED provide one level of information once its modularity technique predicts only the concept of Event. In this sense, cover all MABUP concepts we need to use CED with another business process modeling language, in this case we choose BPMN.

In **Figure41**, the packages in gray (ServiceLevel, OrganizationalLevel, TechnologicalLevel and OperationalLevel) gather the MABUPMM metaclasses. In turn, the white ones address the metaclasses that may be used (e.g. CommunicativeEventDiagram (RUIZ; ESPAÑA; GONZÁLEZ, 2010) and the BPMN2 (OMG, 2010b) packages) in order to employ (i.e. the Adapter package) our metamodel by using existing modeling proposals. Note that we are using the "Adapter" design pattern (GAMMA et al., 1995), which provides a generic interface that allows our metamodel to be adapted and used by other business process modeling proposals.

In our evaluation, we do not capture all context variations required for self-configuration, self-healing and self-protection. Hence, this proposal focuses on the self-optimization feature of the "Declarant submits a declaration" critical communicative event.

Using Communicative Event Diagram to Indicate Organizational Level

According of the principles of Communication Analysis, Communicative Event Diagram provides a notation to specify the Organizational Level of a business process (ESPANA et al., 2009). In this notation, there is a special kind of event, named communicative event that triggers an activity that receives an incoming message, processes it and provides an output. Hence, it represents the organizational behavior resulting from a given change in world (subject system), intended to account for that change by gathering information about it.

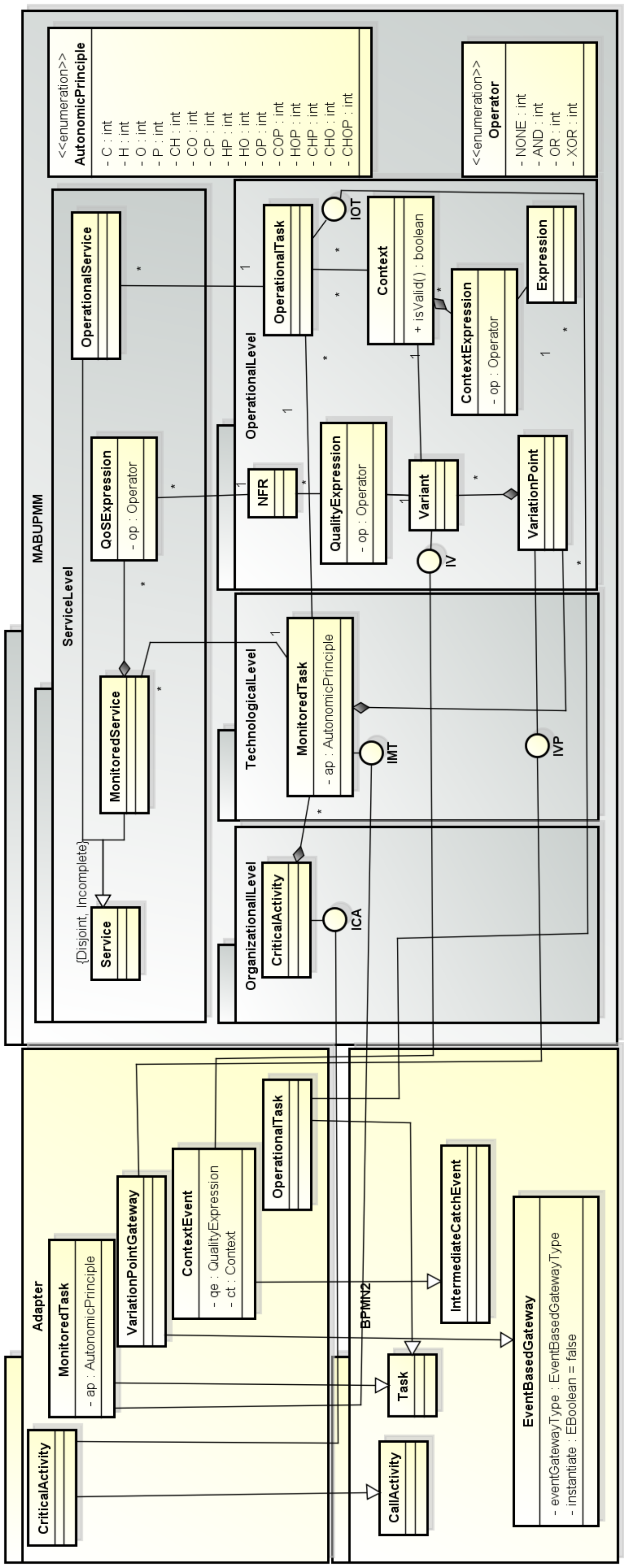


Figure 41 - Extending CED and BPMN with MABUPMM

We introduced the concept of Critical Event, a special kind of communicative event which is critical activity according MABUP. Critical events must be refined to provide information about its behavior and indicate the sub-activities that are monitored according to autonomic features.

In this sense, **Figure 42** presents the Organizational Level of Business Process of CAGED expressed in a Communicative Event Diagram, where we highlight the critical event "CAG2 - Declarant submits declaration". It is an activity that has as input all CAGED movements and as an output the receipt. Note that it is critical to the success of CAGED process.

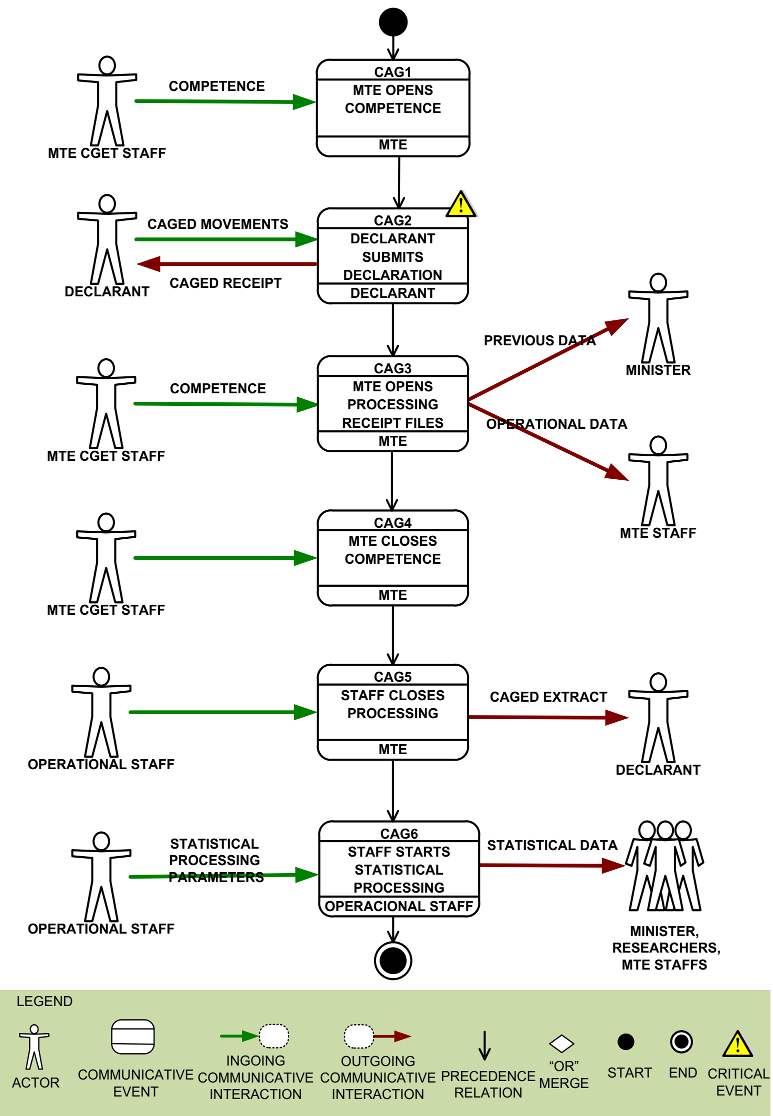


Figure 42 - Organizational Level of Business Process of CAGED using Communication Analysis Principles

Although CED be a proper notation to express the organizational level of business processes, it was not effective for modeling the other levels of MABUP approach. For this reason, it was not adequate enough to incorporate autonomic features but proved to be effective to supply a clear level of modularity.

Appendix B - Tool Support

In the previous sections, we presented the business process managements systems and their benefits as well as MABUP approach. In this appendix, we detail a BPMS and extended BPMN in order to execute MABUP models. Here, we used Activiti framework once it is an environment SOA readiness with workflow engine capabilities. The first section of this chapter provides an overview of the tool, presenting its requirements and user interface. The second section describes the mechanisms we extended.

Activiti Framework: BPM Solution

The Activiti project was started in 2010 by Tom Baeyens and Joram Barrez, the former founder and the core developer of jBPM[2] (JBoss BPM), respectively. The goal of the Activiti project is to build a rock-solid open source BPMN 2.0 process engine. Activiti is funded by Alfresco (known for its open source document management system of the same name; see www.alfresco.com), but Activiti acts as an independent, open source project. Alfresco uses a process engine to support features such as a review and approval process for documents, which means that the document has to be approved by one user or a group of users. For this kind of functionality, Activiti is integrated into the Alfresco system to provide the necessary process and workflow engine capabilities.

Activiti provides a complete BPM solution, starting with the Activiti Designer to draw business processes using BPMN. The XML output of the Activiti Designer is deployed to the Activiti Engine that runs the process definition. The Activiti Engine executes automated steps, like calling a web service, as well as manual steps that involve people and web forms.

[2] jBPM was used in the past instead of Activiti to provide this process and workflow functionality. jBPM is still included in Alfresco, but it may be deprecated at some point in time

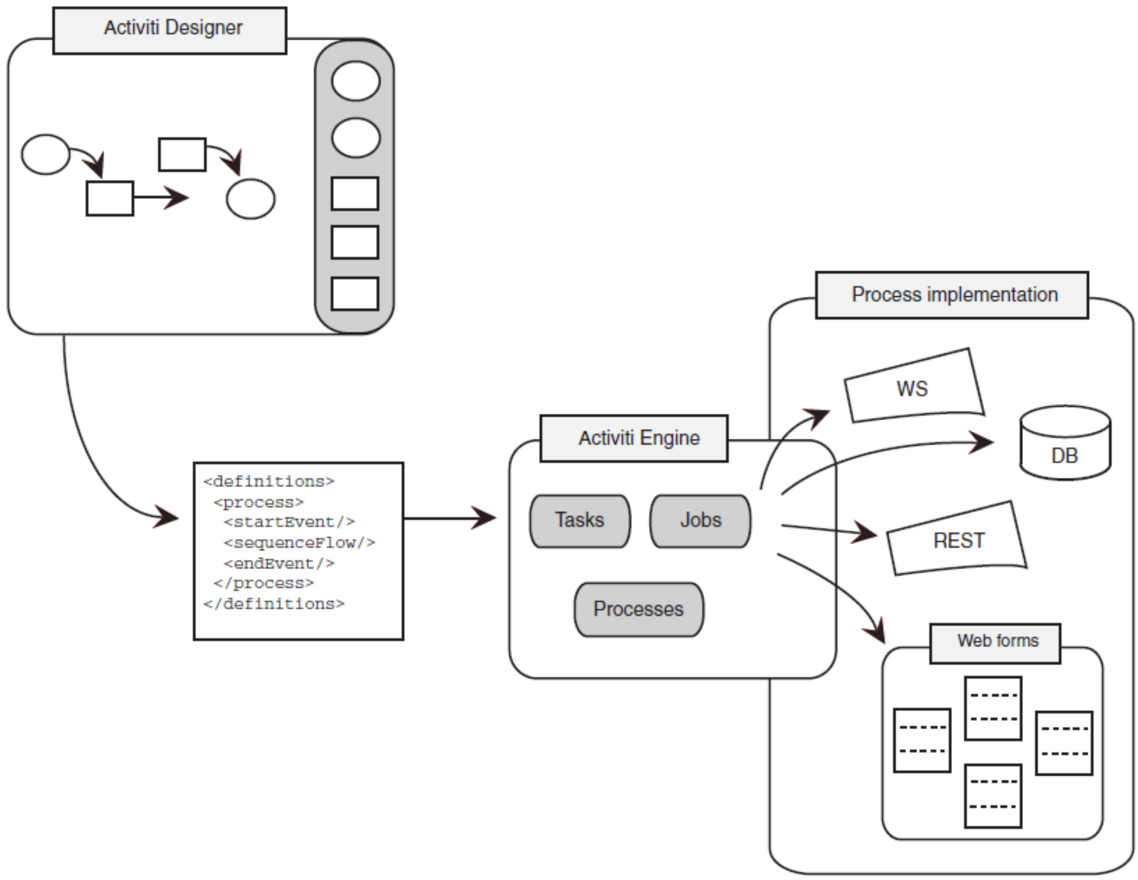


Figure 43 - Activiti Solution (RADEMAKERS; BARREZ, 2012)

The core component of the Activiti framework is the process engine. The process engine provides the core capabilities to execute Business Process Model and Notation (BPMN) 2.0 processes and create new workflow tasks, among other things. The Activiti project contains a couple of tools in addition to the Activiti Engine.

Figure 44 presents an overview of Activiti tools, which can be summarized above: Component name Short description

- **Activiti Engine:** The core component of the Activiti tool stack that performs the process engine functions, such as executing BPMN 2.0 business processes and creating workflow tasks.

- **Activiti Modeler:** A web-based modeling environment for creating BPMN 2.0-compliant business process diagrams. This component is donated by Signavio, which also provides a commercial modeling tool, named the Signavio Process Editor.

- **Activiti Designer**: An Eclipse plugin that can be used to design BPMN 2.0-compliant business processes with the addition of Activiti extensions, such as a

Java service task and execution listeners. It is also possible to do unit test processes, import BPMN 2.0 processes, and create deployment artifacts.

- **Activiti Explorer:** A web application that can be used for a wide range of functions in conjunction with the Activiti Engine. It is possible, for example, start new process instances and get a list of tasks assigned to you. In addition, it is possible to perform simple process management tasks, like deploying new processes and retrieving the process instance status.
- **Activiti REST**: A web application that provides a REST interface on top of the Activiti Engine. In the default installation, the Activiti REST application is the entry point to the Activiti Engine.
- In the center, the Activiti process engine, and on the right and left sides, the accompanying modeling, design, and management tools. The grayed-out components are add-ons to the core Activiti framework

In the center, the Activiti process engine, and on the right and left sides, the accompanying modeling, design, and management tools. The grayed-out components are add-ons to the core Activiti framework

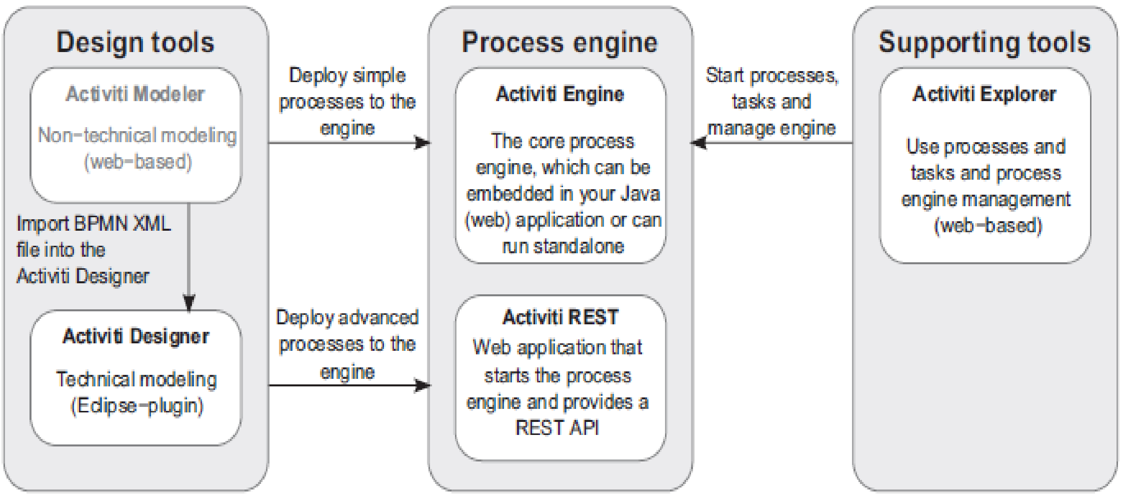


Figure 44 - Overview of Activiti tool stack

The basics of the Activiti Engine

Activiti is a BPMN 2.0 process-engine framework that implements the BPMN 2.0 specification. It is able to deploy process definitions, start new process instances, execute user tasks, and perform other BPMN 2.0 functions.

But at its core, the Activiti Engine is a state machine. A BPMN 2.0 process definition consists of elements like events, tasks, and gateways that are wired together via sequence flows (think of arrows). When such a process definition is deployed on the process engine and a new process instance is started, the BPMN 2.0 elements are executed one by one. This process execution is similar to a state machine, where there is an active state and, based on conditions, the state execution progresses to another state via transitions (think again of arrows).

Figure 45 presents an abstract figure of a state machine and how it is implemented in the Activiti Engine.

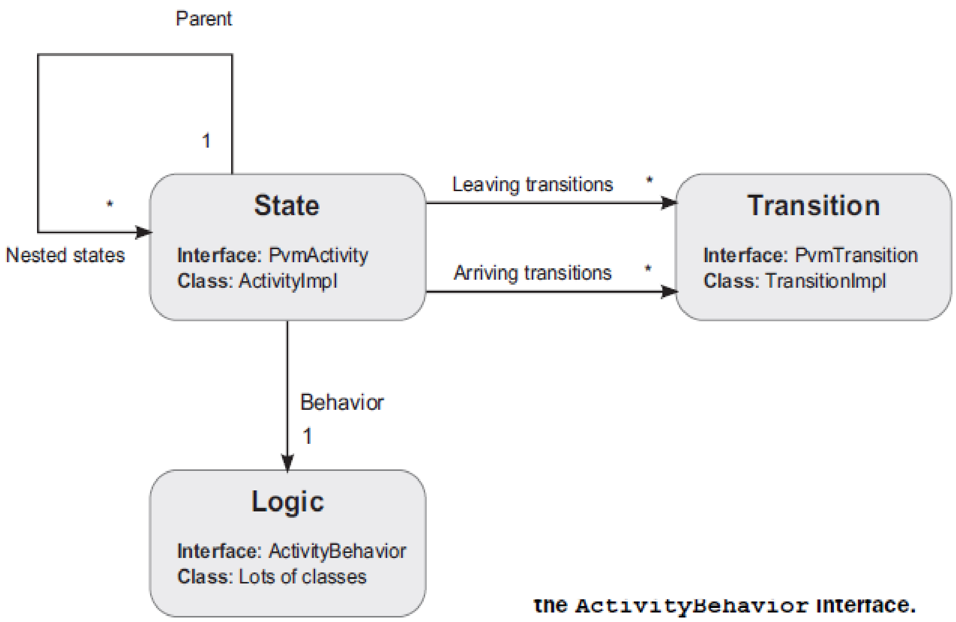


Figure 45 - Abstract overview of state machine in Activiti Engine

In the Activiti Engine, most BPMN 2.0 elements are implemented as a state. They're connected with leaving and arriving transitions, which are called sequence flows in BPMN 2.0. Every state or corresponding BPMN 2.0 element can have attached a piece of logic that will be executed when the process instance enters the state. In **Figure 45**, you can also look up the interface and implementing class that are used in the Activiti Engine. As you can see, the logic interface ActivityBehavior is implemented by a lot of classes. That's because the logic of a BPMN 2.0 element is implemented there.

Extending Activiti Modeler with MABUP Concepts

Initially, we analyzed in **Table 33** the relationship between MABUP concepts and BPMN2.0 concepts. In **Figure 46,** the packages in gray (OrganizationalLevel, TechnologicalLevel, OperationalLevel and ServiceLevel) gather the MABUPMM metaclasses. In turn, the white ones address the metaclasses that may be used (ie. BPMN2 (OMG, 2010b) packages) in order to employ (i.e. the Adapter package) our metamodel by using existing modeling proposals. Note that we are using the "Adapter" design pattern (GAMMA et al., 1995), which provides a generic interface that allows our metamodel to be adapted and used by other business process modeling proposals. (FIGUEIREDO; CASTRO; OLIVEIRA, 2015)

Table 33 - Extension of MABUP concepts in BPMN

Concept	***BPMN Element***	***Description***
Organizational Level	**Process**	**Process model with a well-defined level that could be immune to technological changes**
Critical Activity	Sub-process (new element: Critical Sub-process)	Mission-critical activity that must be refined to provide information about its behavior and indicate the tasks that must be monitored.
Technological Level	**Sub-Process Model**	**Technological Level represents the sub-division of a critical sub-process processing to indicate important aspects that can impact software adaptation**
Monitored Task	Service Task (new element: Monitored Service Task)	Task that must be monitored according autonomic principles.
Autonomic Principle	Service Task attribute	Indicates autonomic principle that must be assured.
Operational Level	**Sub-Process Model**	**Operational level indicates the operational knowledge required to manage the process. Business analysts define the contextualization of the monitored task using their knowledge about the business domain to identify information that can affect the process**

Concept	*BPMN Element*	*Description*
		and express the operational knowledge to manage it.
Variation Point	Event-based exclusive gateway (new element: Variation Point)	Contextual information related to environment variation that affects the system, ie. contextualization for alternative ways of how the adaptation actions are selected to respond to changes.
Variant	Intermediate Catch Event	Alternative associated with the variation point.
Quality Expression	Intermediate Event attribute	Expression related to the metric of quality attributes that must be checked in the variant.
Context	Intermediate Catch Event attribute	Set of operational environment situations that occur and affect the business process execution.
Operational Task	Service Task (new element: Operational Service Task)	Task that express autonomic action in the systems to assure the optimal state of the system.
Service Level	**No Element**	
Service		System operation that executes a process (ie. Process, activity or task) independent if it is autonomic or not.
Monitored Service		Service that has a specific SLO to be assured.
Operational Service		Service that executes operational task to recover the system to a optimal state.
QoS Expression		Expression related to the metric of quality attibutes contained in the SLO and provided by service monitoring

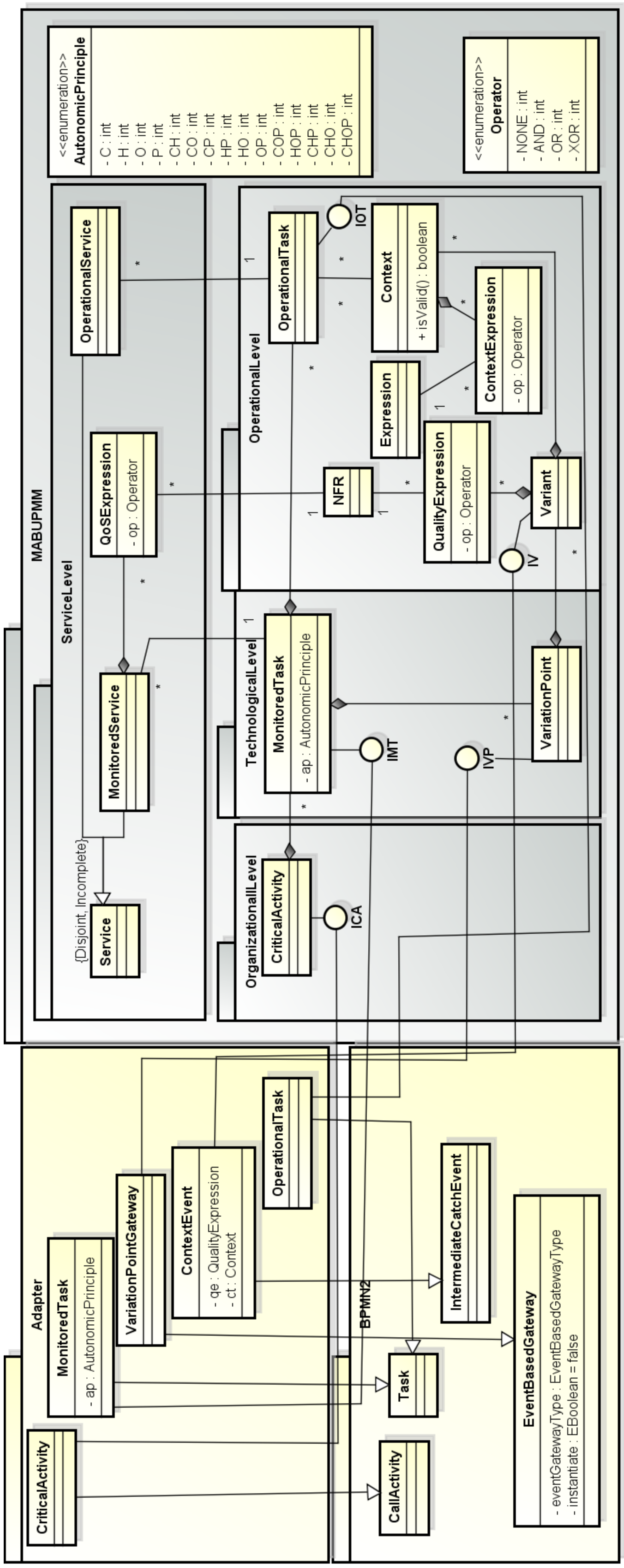
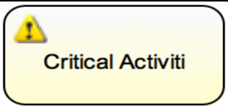
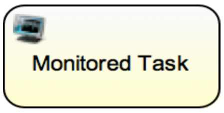
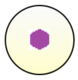
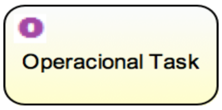


Figure 46 - Extension of BPMN2 to incorporate MABUP concepts

All concepts were represented as described in **Table 34**. **Figure 47** shows activity with MABUP pallet, which contain all MABUP elements and attributes to be used in the modeling activity.

Table 34 - MABUP Elements and its representation in BPMN

MABUP Element	Element Representation	Attributes
Critical Activity		
Monitored Task		AutonomicPrinciples: Enumeration
Variation Point		Name: String
Context		Context Expression: Expression [] NFR: Expression []
Operational Task		

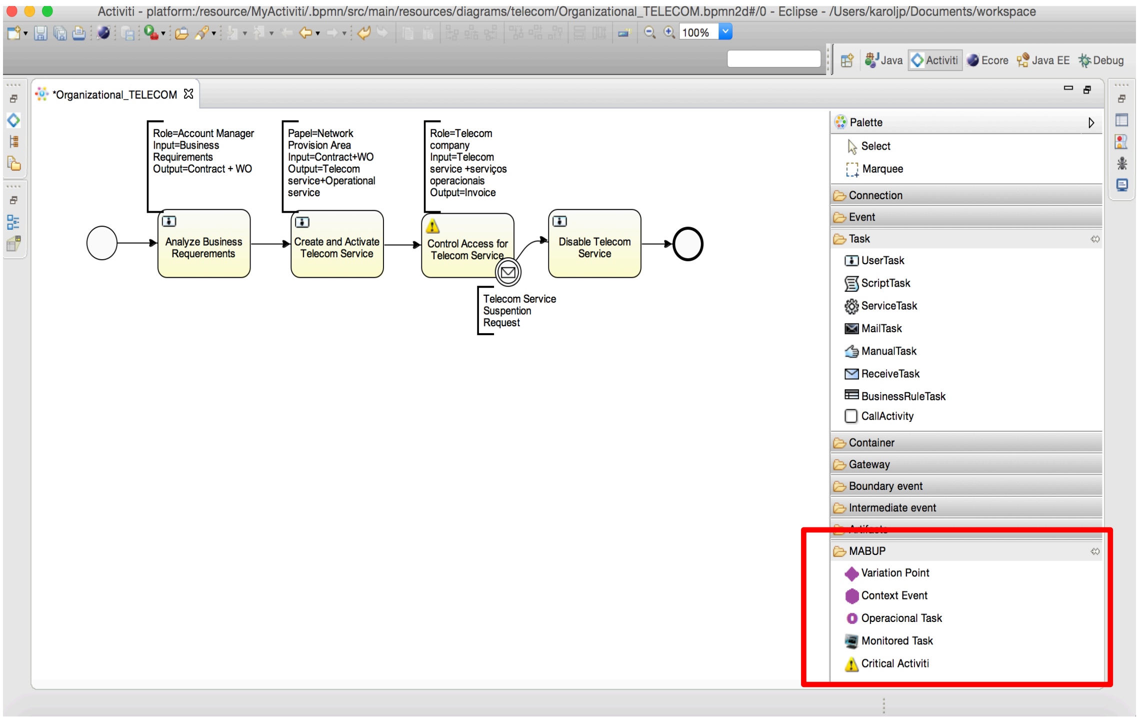


Figure 47 - Overview of Activiti with MABUP pallet

In addition to the design tools, you can also use Activiti Explorer to interact with the deployed business processes. For example, you can get a list of tasks that are already assigned to you. You can also start a new process instance and look at the status of that newly created process instance in a graphical diagram. With your fields setup and your extension applied to Designer, users can configure the properties of the service task when modelling a process. In most cases, you will want to use these user-configured properties when the process is executed by Activiti.

To make a deploy process definitions and task forms on the Activiti Engine you need a BAR file containing the process definition BPMN 2.0 XML file and optionally task forms and an image of the process that can be viewed in the Activiti Explorer. When you have finished your process implementation create deployment artifacts and generate bar file as can be seen in **Figure 48**.

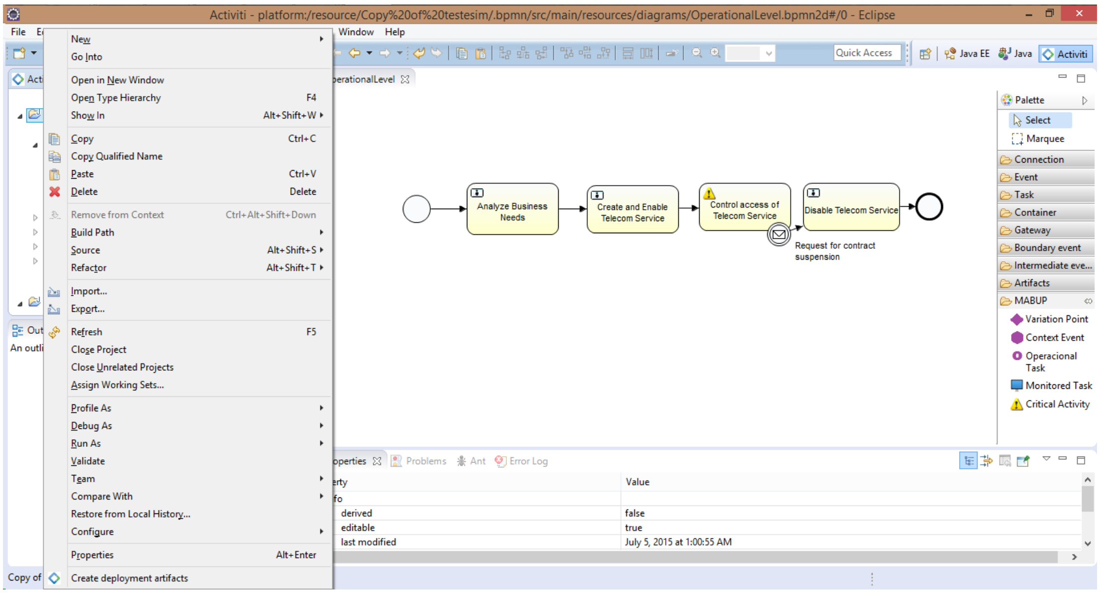


Figure 48 - Example of creating the deployment artifacts

When the Organizational Level archive is uploaded to the Activiti engine, it is automatically deployed, versioned, and available in the Activiti Explorer for starting a new process instance as presented in **Figure 49**.

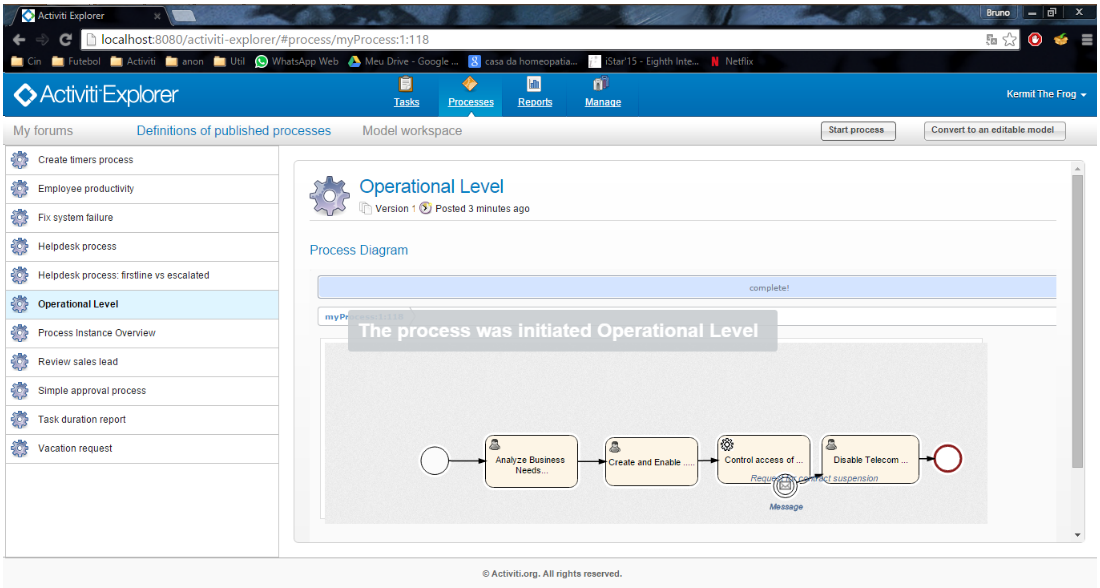


Figure 49 - Example of deploying Organizational Level in Activiti Explorer

The concepts defined in the MABUP approach was implemented for the running process on the engine Activiti and test it with the Activiti Explorer application. In other words, it is possible from the abstract code model for concrete, but the implementation details of the extensions and the MABUP approach in Activiti will be better described in further work.

Appendix C - Survey of Participant Profile

Survey of Participant Profile

The following questionnaire aims to provide information on your background about business modeling and software engineering. Your answers will not affect other activities of this experiment; it will just provide a context for the interpretation of results. Feel free to write in the margins to explain your answers, if necessary.

Name*

Are you a student of which course?

- Graduation
- Master
- Doctorate

What is the name of your course?

Do you have professional experience in software engineering or business analysis?

- No
- Yes

If the previous answer was 'YES', which activities?

Do you have used some language for business modeling before this discipline?

- UML
- BPMN
- BPEL
- Communication Analysis (CED)
- Others

I am proficient in business process modeling.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

I am proficient in performing analysis and system design.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

I am proficient in another way of business representation than business process model.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Appendix D - Case Study – Telecom Services Management (MABUP group)

Experiment Execution

Group: **MABUP**

Date: 04/29/2015

Case Study - Telecommunications (Telecom) Services Management

1) Model the business process of Telecom Services Management.

The telecom services management process supports the operation of telecom services hired by corporate clients to meet their business needs. The process begins when the business needs are analyzed by an **account manager** that formalizes a telecom service contract, which predicts a service level agreement (SLA)[3]; and produces a work order (WO) to be implemented. The **network provisioning area** examines the WO and the SLA to create the telecommunications service as specified for the client, activates it considering the defined rules and implements operational services that are necessary to manage its operating environment. After activation of the telecom service, subscribers are able to use it, thus, the telecommunications company begin to control their access until receive a request to suspend the service contract.

The activity that controls access to telecom service is critical to the success of the telecom service management process. The failure of this activity can cause an exponential financial loss. The objective of this activity is to receive an access request, perform the charges, inspect service access and upgrade the request. Thus, the access control is started when the **network area** receives an access request to telecom service that can be <u>start request</u>, <u>update request</u> or <u>end request</u>. From this, the **mediation area** can perform three different actions:

(i) If the request is start access, verify if the subscriber is allowed to access the requested service. If so, the billing/charging area starts to record the beginning of the data traffic; otherwise, the area networks denies the service access and ends the process;

(ii) If the request is update access, verify if the subscriber has access quotas to continue service access through online billing (default route) or an alternative billing to allow or deny access [only if the online billing fails, the system can perform the alternate billing (billing hot) until the system recovers from the failure, once this method is more expensive and susceptible to problems]. If **mediation** allows access, the **billing/charging area** updates the access of data traffic, but the **networks area** denies the service and ends the process.

[3] SLA - It is a detailed document that defines the standards of a service level and the relationship between two parties: requester and executor.

(iii)If the request is end access, end the traffic of telecom service. Later, the **billing/charging area** records to the closure of traffic and ends the process.

By having access records held by **billing/charging area**, **mediation area** allows the service traffic. The **networks area**, in turn, allows one of the different types of available services to be accessed depending on the request, ie. mobile data service, fixed internet services, fixed voice long distance service and mobile voice service for long distance. When the quota is being used, the traffic is inspected to request the continuity of the service access by sending a new update access request for a new charging.

There is a requirement ruled in the customer contract that the telecom services cannot have failures, specifically, no loss of data traveling on the network. For example, the task "Allow mobile data service access" should provide self-healing, ie, it should have tasks to conduct operational adjustment actions if there is network data loss. In order to address the data loss deviations attribute, that is associated with the non-functional requirements of *performance,* three different actions can be activated depending on the context:

(i) Increase of 1 db the laser power of the transmitter (if the data loss is less than 1%, there is an expected increase in the loss rate in the last three cycles of monitoring and reception power was modified);
(ii) Reset the transmitter board (if the data loss is less than 1%, there is an expected increase in the loss rate in the last three cycles of monitoring and reception power was not modified); and
(iii)Reset network gateway (if the data loss is less than 1%, the response time is higher than 110 ms and there is an expected increase in the loss rate in the last three cycles of monitoring).

a) Model organizational level of the business process (task 1 MABUP approach)

Identify the roles and artifacts according of MABUP criteria for organizational level.

Identify the critical activities according this business description.

b) Model technology level of the business process (task 2 MABUP approach)

Define the tasks that should be performed and monitored according to the autonomic features.

Define how systems should to be adapted in case of any deviation of quality attributes.

In the example, it is necessary to ensure self-healing in the case of data loss.

c) Model operational business process (step 3 of MABUP approach)

Define quality and context attributes that should be monitored according autonomic features. These attributes must be set in the properties of elements type named Context of MABUP.

Define how systems should be adapted in case of any deviation of quality attributes.

d) Model service level of the business process (step 4 of MABUP approach)

Define which tasks should have the support of services through annotations in the model.

Appendix E - Case Study – Telecom Services Management (Control group)

Experiment Execution

Group: **Control**

Date: 04/29/2015

Case Study - Telecommunications (Telecom) Services Management

1) Model the business process of Telecom Services Management.

The telecom services management process supports the operation of telecom services hired by corporate clients to meet their business needs. The process begins when the business needs are analyzed by an **account manager** that formalizes a telecom service contract, which predicts a service level agreement (SLA)[4]; and produces a work order (WO) to be implemented. The **network provisioning area** examines the WO and the SLA to create the telecommunications service as specified for the client, activates it considering the defined rules and implements operational services that are necessary to manage its operating environment. After activation of the telecom service, subscribers are able to use it, thus, the telecommunications company begin to control their access until receive a request to suspend the service contract.

The activity that controls access to telecom service is critical to the success of the telecom service management process. The failure of this activity can cause an exponential financial loss. The objective of this activity is to receive an access request, perform the charges, inspect service access and upgrade the request. Thus, the access control is started when the **network area** receives an access request to telecom service that can be start request, update request or end request. From this, the **mediation area** can perform three different actions:

(iv) If the request is start access, verify if the subscriber is allowed to access the requested service. If so, the billing/charging area starts to record the beginning of the data traffic; otherwise, the area networks denies the service access and ends the process;

(v) If the request is update access, verify if the subscriber has access quotas to continue service access through online billing (default route) or an alternative billing to allow or deny access [only if the online billing fails, the system can perform the alternate billing (billing hot) until the system recovers from the failure, once this method is more expensive and susceptible to problems]. If **mediation** allows access, the **billing/charging area** updates the access of data traffic, but the **networks area** denies the service and ends the process.

---

[4] SLA - It is a detailed document that defines the standards of a service level and the relationship between two parties: requester and executor.

(vi) If the request is end access, end the traffic of telecom service. Later, the **billing/charging area** records to the closure of traffic and ends the process.

By having access records held by **billing/charging area**, **mediation area** allows the service traffic. The **networks area**, in turn, allows one of the different types of available services to be accessed depending on the request, ie. mobile data service, fixed internet services, fixed voice long distance service and mobile voice service for long distance. When the quota is being used, the traffic is inspected to request the continuity of the service access by sending a new update access request for a new charging.

There is a requirement ruled in the customer contract that the telecom services cannot have failures, specifically, no loss of data traveling on the network. For example, the task "Allow mobile data service access" should provide self-healing, ie, it should have tasks to conduct operational adjustment actions if there is network data loss. In order to address the data loss deviations attribute, that is associated with the non-functional requirements of *performance,* three different actions can be activated depending on the context:

(iv) Increase of 1 db the laser power of the transmitter (if the data loss is less than 1%, there is an expected increase in the loss rate in the last three cycles of monitoring and reception power was modified);
(v) Reset the transmitter board (if the data loss is less than 1%, there is an expected increase in the loss rate in the last three cycles of monitoring and reception power was not modified); and
(vi) Reset network gateway (if the data loss is less than 1%, the response time is higher than 110 ms and there is an expected increase in the loss rate in the last three cycles of monitoring).

Appendix F - Post-Experiment Questionnaire (MABUP group)

Post experiment questionnaire - MABUP Group

For each of the following statements, indicate strongly agree, agree, neither agree nor disagree, disagree or strongly disagree

What the experiment finish time *?

What's your name? *

Step1 (Define Organizational Level of Business Process) is understandable?

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Step 2 (Define Technological Level of Business Process) is understandable?

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Step 3 (Define Operational Level of Business Process) is understandable?

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree

- Strongly Disagree

Step 4 (Express services of automated tasks - monitored and operational) is understandable?

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Use the MABUP modeling technique is easy?

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Would you like to add a comment about the process?

Appendix G - Post-Experiment Questionnaire (Control group)

Post experiment questionnaire - Control Group

For each of the following statements, indicate strongly agree, agree, neither agree nor disagree, disagree or strongly disagree

What the experiment finish time *?

What's your name? *

The modeling of the business process using BPMN is easy

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

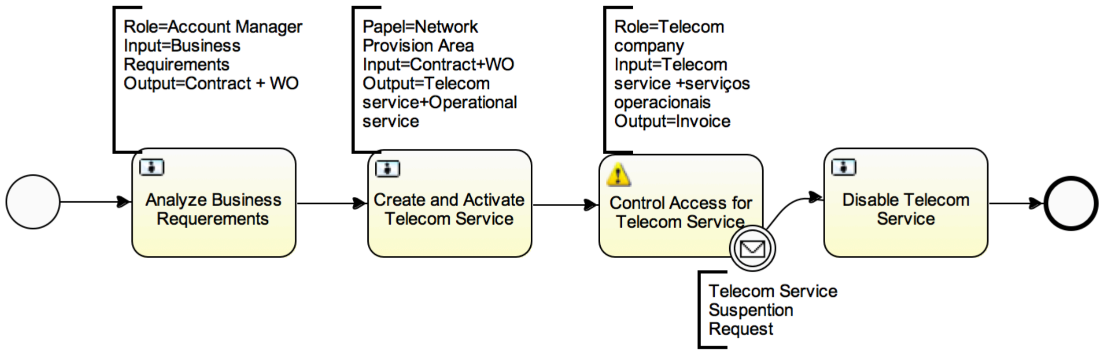
Would you like to add a comment about the process?

Appendix H - Reference Model of Telecom Service Management to MABUP group

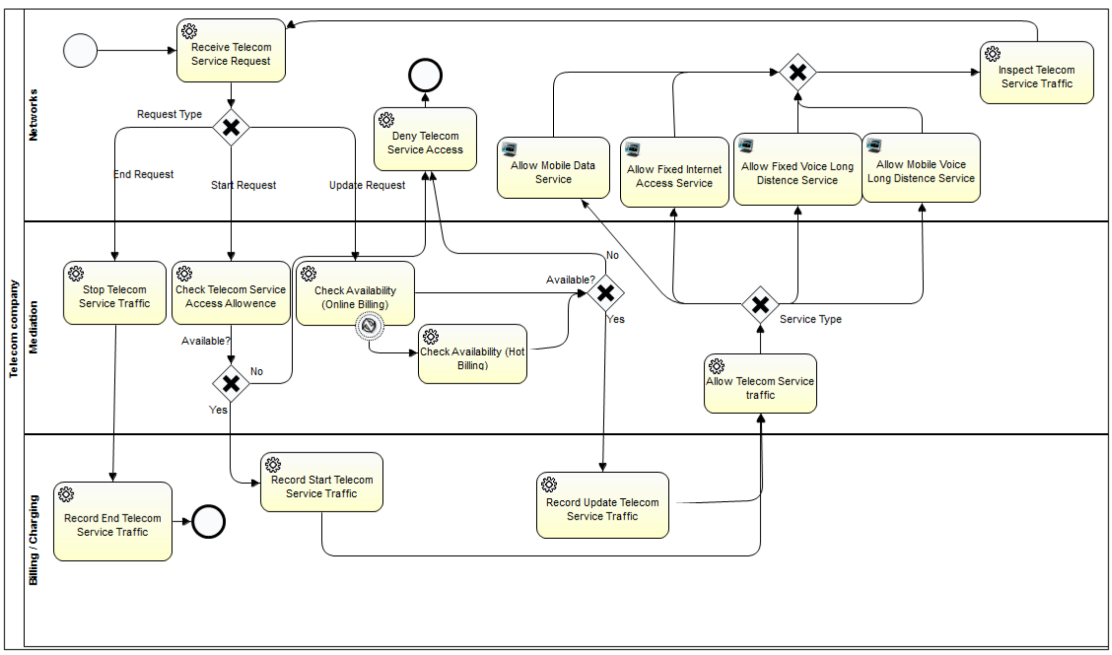
Reference Model of Telecom Service Management (MABUP approach)

Three levels defined in MABUP approach were modeled by an expert research to be compared with the models created by subjects that participated of the controlled experiment.

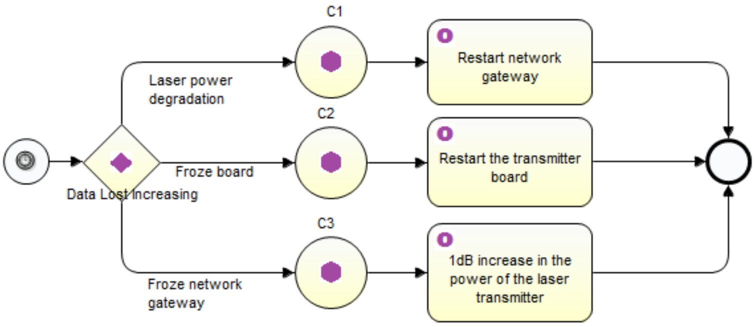
Step 1 - Organizational Level



Step 2 – Technological Level



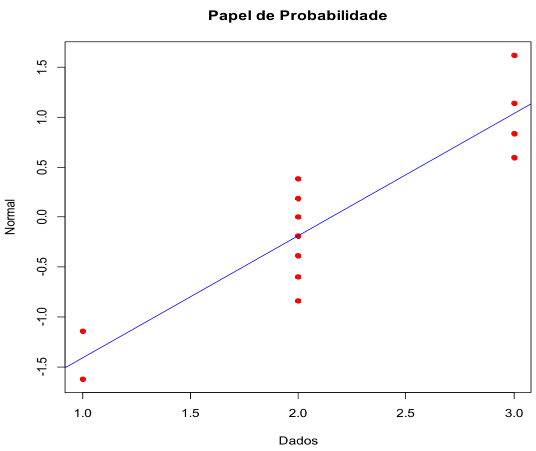
Step 3 – Operational Level

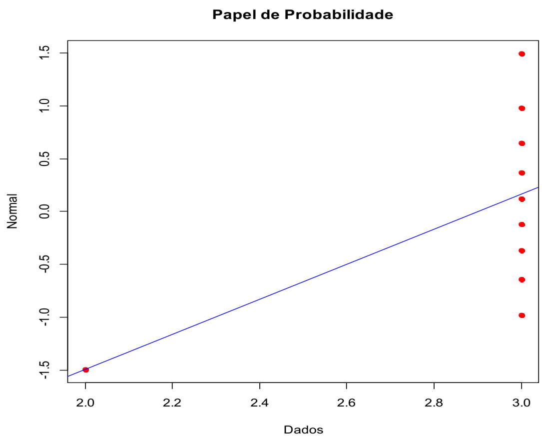


Appendix I - Normality Test and F Test Results

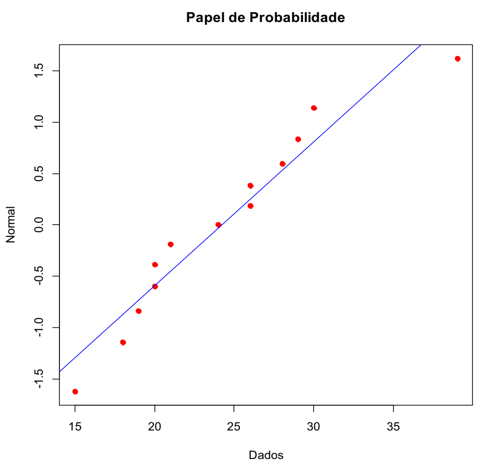
Normality Test and F test to Quantitative Metrics

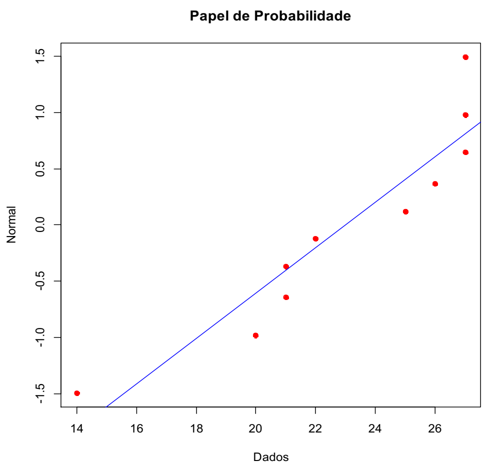
Normality Test of NLM			
Control		MABUP	
Statistic: Anderson-Darling	1,167086908	**Statistic: Anderson-Darling**	3,208056813
P-value	0,002987722	**P-value**	8,55956E-09



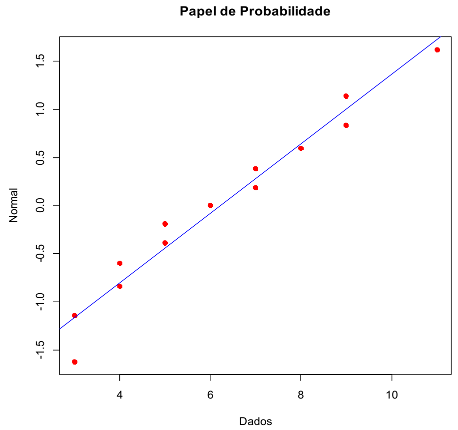


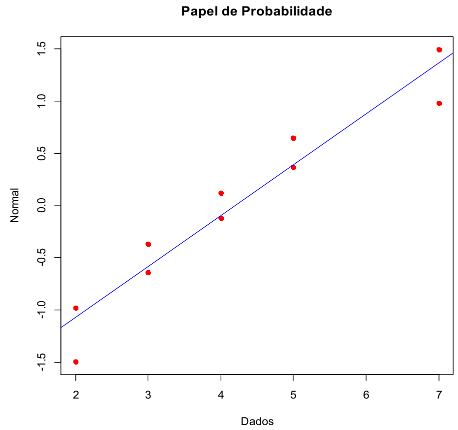
Normality Test of NTM			
Control		MABUP	
Statistic: Anderson-Darling	0,309948831	**Statistic: Anderson-Darling**	0,522610619
P-value	0,511523056	**P-value**	0,136617048



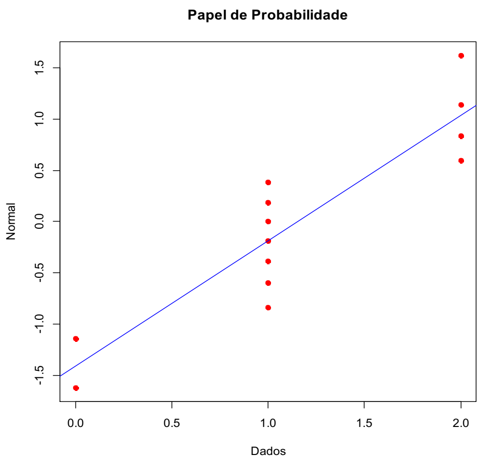


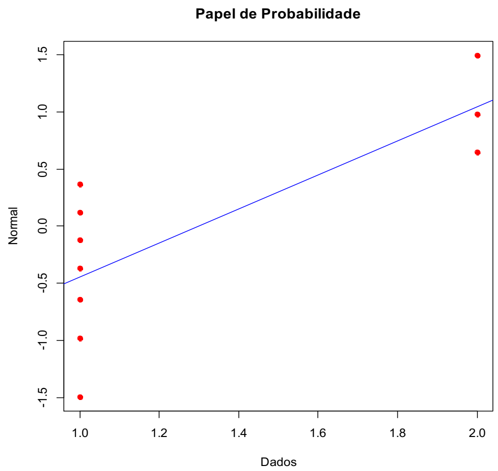
Normality Test of NMM			
Control		MABUP	
Statistic: Anderson-Darling	0,269589019	**Statistic: Anderson-Darling**	0,352588417
P-value	0,616429301	**P-value**	0,388768479



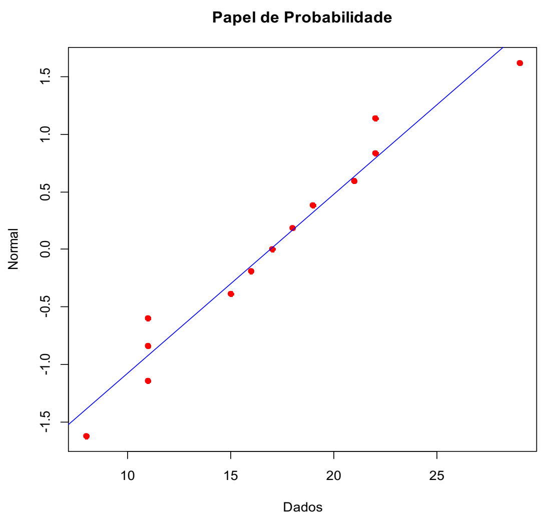


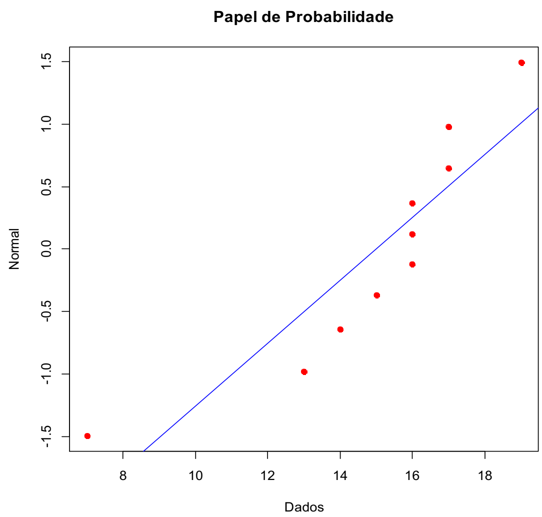
Normality Test of NSM			
Control		MABUP	
Statistic: Anderson-Darling	1,167086908	**Statistic: Anderson-Darling**	2,033125085
P-value	0,002987722	**P-value**	1,17401E-05



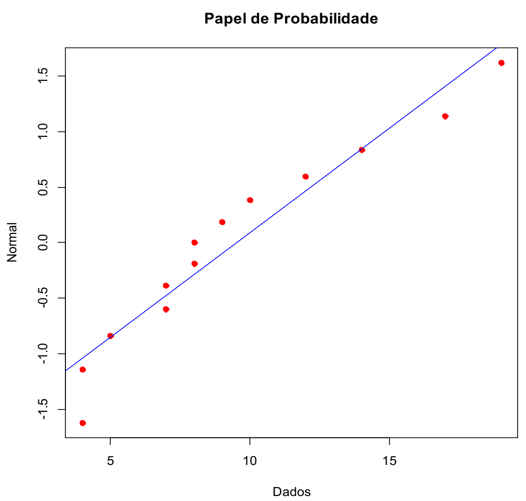


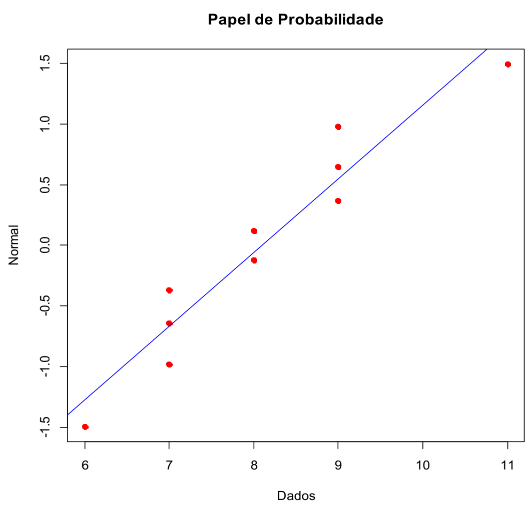
Normality Test of NTL			
Control		MABUP	
Statistic: Anderson-Darling	0,248466049	**Statistic: Anderson-Darling**	0,710269064
P-value	0,690888438	**P-value**	0,043054908



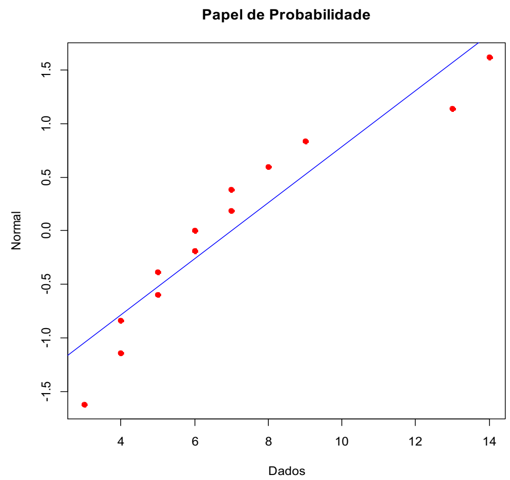


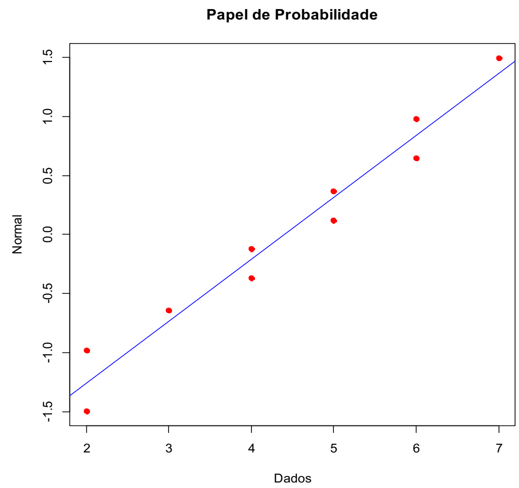
Normality Test of NEM			
Control		MABUP	
Statistic: Anderson-Darling	0,421588265	**Statistic: Anderson-Darling**	0,371163083
P-value	0,273702208	**P-value**	0,348419656





Normality Test of NGM			
Control		MABUP	
Statistic: Anderson-Darling	0,609848001	**Statistic: Anderson-Darling**	0,252281406
P-value	0,088283403	**P-value**	0,654807697





F TEST - TWO VARIANCES

NLM - Control	NLM - MABUP

Information	***Value***
F Statistic	4,743589744
Degrees of freedom (numerator)	12
Degrees of freedom (denominator)	9
P-Value	0,025760445
Sample standard deviation (sample 1)	0,688737232
Sample standard deviation (sample 2)	0,316227766
Confidence Interval for the Ratio of Variances	95%
Inferior Limit	1,226297715
Superior Limit	16,29824215

F TEST - TWO VARIANCES

NTM - Control	*NTM - MABUP*

Information	***Value***
F Statistic	2,307692308
Degrees of freedom (numerator)	12
Degrees of freedom (denominator)	9
P-Value	0,215871886
Sample standard deviation (sample 1)	6,405126152
Sample standard deviation (sample 2)	4,216370214
Confidence Interval for the Ratio of Variances	95%
Inferior Limit	0,596577267
Superior Limit	7,928874558

F TEST - TWO VARIANCES

NMM - Control	*NMM - MABUP*

Information	***Value***
F Statistic	1,933471933
Degrees of freedom (numerator)	12
Degrees of freedom (denominator)	9
P-Value	0,328000752
Sample standard deviation (sample 1)	2,521700688
Sample standard deviation (sample 2)	1,813529401
Confidence Interval for the Ratio of Variances	95%
Inferior Limit	0,499835007
Superior Limit	6,643111116

F TEST - TWO VARIANCES

NSM - Control	*NSM - MABUP*

Information	***Value***
F Statistic	2,032967033
Degrees of freedom (numerator)	12
Degrees of freedom (denominator)	9
P-Value	0,292664021
Sample standard deviation (sample 1)	0,688737232
Sample standard deviation (sample 2)	0,483045892
Confidence Interval for the Ratio of Variances	95%
Inferior Limit	0,525556164
Superior Limit	6,98496092

F TEST - TWO VARIANCES

NTL - Control	*NTL - MABUP*

Information	***Value***
F Statistic	3,194711538
Degrees of freedom (numerator)	12
Degrees of freedom (denominator)	9
P-Value	0,089349696
Sample standard deviation (sample 1)	5,837544268
Sample standard deviation (sample 2)	3,265986324
Confidence Interval for the Ratio of Variances	95%
Inferior Limit	0,825886654
Superior Limit	10,97653572

F TEST - TWO VARIANCES

NEM - Control	*NEM - MABUP*

Information	***Value***
F Statistic	10,76312576
Degrees of freedom (numerator)	12
Degrees of freedom (denominator)	9
P-Value	0,001262627
Sample standard deviation (sample 1)	4,754215403
Sample standard deviation (sample 2)	1,449137675
Confidence Interval for the Ratio of Variances	95%
Inferior Limit	2,782448998
Superior Limit	36,98043875

F TEST - TWO VARIANCES

NGM- Control	*NGM - MABUP*

Information	***Value***
F Statistic	3,806818182
Degrees of freedom (numerator)	12
Degrees of freedom (denominator)	9
P-Value	0,052571949
Sample standard deviation (sample 1)	3,341656276
Sample standard deviation (sample 2)	1,712697677
Confidence Interval for the Ratio of Variances	95%
Inferior Limit	0,984126514
Superior Limit	13,07963966

F TEST - TWO VARIANCES

MDT- Control	*MDT - MABUP*

Information	***Value***
F Statistic	2,48909859
Degrees of freedom (numerator)	12
Degrees of freedom (denominator)	9
P-Value	0,17805555
Sample standard deviation (sample 1)	0,007760939
Sample standard deviation (sample 2)	0,004919186
Confidence Interval for the Ratio of Variances	95%
Inferior Limit	0,643473841
Superior Limit	8,552158544