ABSTRACT
The Academy Management System in Instituto Superior Técnico (Fénix) has grown quickly over the last four years. This growth was concerned with functionality development, without regarding to their context on the business processes where they are included. The challenge of this work is to find out if there is a possibility of conceiving a solution to combine and articulate those functionalities in a business process approach. Using enterprise architecture methodologies to define the organizational context and the business processes, the aim is to find out, after the business analysis, some commonalities between business and implemented technology. After finding the contact points between business and technology, the evaluation of the possibility of having a parallel system to fulfill the gap is studied.

A concrete example of this misalignment is the process of attribution and execution of the Master’s Thesis. Applying well known methodologies, the formalization of the business process showed that it was possible to connect the data, on the Fénix System, to a process definition, deployed in a Business Process Management System. However, not all cases are so simple to achieve the solution, when leading with insufficient information and high complexity, either from the business or from the technology side.

Keywords

1. INTRODUCTION
Every graduation institution has a set of regulated procedures related with its activity. The need for efficiency on execution, as well as the amount of activities required in shorter timings has increased over the last years. The main cause for that efficiency increase was the growing technological ability to automate time consuming procedures. The problem is that the IT advance has not been being accomplished with the procedural regulations. On one hand we have applications to support single procedural tasks; on the other hand they are not orchestrated in a perspective of a process. The problem of this approach is the impossibility to set a global state of the process execution. As consequences of that, it is hard to coordinate effort and audit the process. In this scenario the aim for efficiency is a hard goal to attain.

Instituto Superior Técnico (IST) is not an exception to the above mentioned problem. In this case, there is an information system, Fénix, which sustains a wide range of tasks in the academy. Due to its purely technological start, the system is agnostic in what concerns to the business processes. The result is that the activities executed on it are not linked to the process where they belong within the system. The process track is done only by the participants.

1.1 Goals
The main goal of this work is to evaluate the possibility of aggregating process vision of the problem with its activities’ implementation, without having to change or rebuild an existing system.

The start point for the study of a solution is based on a concrete problem, the Educational and Research departments on IST. Each of these departments shares a set of previously regulated procedures. One of these procedures, the Master’s Thesis process, is the chosen case to set up a detailed analysis and try to conceive a solution that deals with the process definition from one side and existing technological support by the other. This process is well documented by the institution and is supported in its main activities by the existing information System.

1.2 Structure
This text is structured in eight chapters:

- On chapter one, an overview of the whole work is done, as well as the goals to achieve and the work structure
- On chapter two the effort is directed to the study of existing knowledge - tools, methodologies and technologies, to explore the problem and the solution
- On chapter three begins the application of the contents from chapter two, by accurately defining the process, as described on the regulation.
- On chapter four, is time to analyze Fénix System. With the previously defined process (business) the aim is to find where and how its activities are supported by the system (technology).
- Having the both sides of the problem, business and technology, on chapter five is presented a possible solution for the lack of business process support by the current system.
- On chapter six, the previously presented solution is partially implemented, for a smaller universe. The limitations of the presented solution are also presented here.
- Chapter seven is where the implemented solution is evaluated, by checking if it solves the problem presented on chapters three and four, either with data changes, scenarios and less frequent cases.
- Finally on chapter eight the whole work is evaluated in terms of a solution for the global problem of fulfilling the gap between business and technology. General conclusions about the used methodology on other contexts are made as well as the future work that can be done.
2. STATE OF THE ART

This chapter is dedicated to the research on existing contents in what concerns to methodologies, frameworks and tools. The studied contents are tightly related with the problem definition/analysis and solution design.

The approach starts with frameworks and methodologies for Enterprise Architectures. Workflow management, process modeling and execution are the next subjects on analysis.

2.1 Enterprise Architectures

Enterprise Architecture (EA) concept was first used by John Zachman, on the framework with the same name. This framework was released in 1987, and reviewed in 1992. On today’s work this framework serves as a base for newer EA frameworks.

The IEEE 1471 standard defines EA as follows [1]:

_The fundamental organization of a system embodied by its components, their relations to each other and to the environment, and the principles guiding its design and evolution._

According to this definition, two different parts arise – the “fundamental organization”, can be viewed as the framework. The “principles to guide design and evolution” are the methodology used to fulfill the framework defined in the first part.

2.1.1 Zachman Framework

First presented on the 80’s, the initial version of the Zachman framework was targeted directly to information system architectures [19], rather than enterprise architectures. The concept of enterprise arrived on 90’s, and now it is called Zachman’s Framework for Enterprise Architecture [14]. The framework is a matrix, where the rows are called perspectives and the columns are dimensions.

Perspectives in Zachman’s Framework correspond to different levels of detail of the EA. The aim for that division is to take into consideration all the participants involved in planning, conception, building, usage and maintenance activities in an organization EA.

On the first version of the framework J. A. Zachman proposed three dimensions [19] - Data, Process and Network. In the latter one, six dimensions exist [14]: Data, Function, Network, People, Time and Motivation. The aim for the existence of dimensions on the framework is the need to provide focusing in distinct variables while maintaining others constant.

2.1.2 Enterprise Architecture Definition

The Enterprise Architecture Definition is the most used model to describe EA. This model divides an EA into four components: Business Architecture, Information Architecture, Application Architecture and finally Technical Architecture. The set of Information and application architectures is called Information System Architecture, as in Figure 1 [12].

**Business architecture** is the result of defining all the business concepts related with the enterprise: Strategy maps, goals, policies, operating models and business processes.

**Information architecture** concerns with data’s physical and logical aspects. It also involves the management of data resources, originated by the data needs from business processes.

**Application Architecture** focuses on application development and implementation, in order to fulfill the requirements stated on the business architecture.

On **technical architecture**, the foundations that support the applications, data and business processes are found. The needed services are identified and planned into this architecture as well as the technical infrastructure.

2.1.3 Enterprise Architecture Planning Methodology

Enterprise Architecture Planning (EAP) [15] was published in 1992. It is concerned with the first two rows from the Zachman Framework (Scope and Business Model).

EAP is a method, a how-to fill an existing framework. As a main feature for EAP is the fact that it focuses on defining data, applications and technology architectures for the overall enterprise instead of designing each architecture for a specific purpose. The process of defining an EA is divided into seven components, organized in layers as shown on Figure 2.

**Application architecture** defines the support for business supporting systems that are currently running is done - application platforms, data and technology are documented.

The **Business Model** defines the most evident data types that need support.

**Application Architecture** defines the support for business architecture requirements, as well as the needs to handle data types.
Technology Architecture states the technology needed to provide the environment for application architecture and data location.

The left to right arrows indicate the sequence of development of these architectures.

The Implementation/Migration Plans component defines the schedule for implementing applications and cost-benefit analysis. It is a complete plan to migrate from as-is architecture (layer 2) to the to-be architecture (layer 3).

A set of deliverables is also set for each step of the EA building using EAP [15].

2.1.4 Evaluation of Enterprise Architectures

A wider study on EA frameworks is done on with the analysis of other EA frameworks and methodologies such as TOGAF (The Open Group Architecture Framework) [11] and IAF [3], from Capgemini. Table 1 compares one to each other and shows the benefits from each one in particular.

<table>
<thead>
<tr>
<th>Framework</th>
<th>Aim</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zachman</td>
<td>Representation</td>
<td>Complexity divided (perspectives). Aspect division (dimensions).</td>
</tr>
<tr>
<td>EA Definition</td>
<td>Representation</td>
<td>Simplicity of the four components. Easy Change Support</td>
</tr>
<tr>
<td>EAP</td>
<td>Methodology</td>
<td>Completeness. Detailed Artifact definition.</td>
</tr>
<tr>
<td>TOGAF</td>
<td>Methodology</td>
<td>Easy usage with different representations. Iterative Development</td>
</tr>
<tr>
<td>IAF</td>
<td>Methodology</td>
<td>Simplicity. New perspectives on EA (governance and security)</td>
</tr>
</tbody>
</table>

2.2 Workflow Management

Workflow management bridges between business activities and its technological support. To make that bridge information and architecture architectures are used, in which the workflow manager will make the orchestration. The role of workflow is to focus on process execution relying on tasks and in the way how people and applications execute them. A definition for workflow is stated by Workflow Management Coalition (WfMC) [18] as:

The computerized facilitation or automation of a business process, in whole or part.

A workflow management system is a software component that receives a business formal description (vision, goals, objectives, processes) and keeps tracking about process execution. The system also assigns functions through people and/or applications. According to the same institution the definition of workflow management system is:

A system that completely defines manages and executes "workflows" through the execution of software whose order of execution is driven by a computer representation of the workflow logic.

2.2.1 Business Process Management

Business Process Management can be viewed as a set of improvements to workflow. Features such as stronger document management, human activity support, stronger application integration mechanisms, allied with reporting and analysis tools form what can be called a Business Process Management System (BPMS). Pyke considers Business Process Management [13]:

A piece of technology that allows us to create a process layer, which provides a level of process abstraction, and removes the processes from the control of the applications.

The process flow is no longer decided by a sequence of interconnected applications, but assured by an upper layer application, that controls the execution of the processes – the BPMS.

2.2.2 Case Handling Paradigm

Case handling [16] is a new paradigm for flexible and knowledge based business processes. Within its data based approach, Case Handling tends to fulfill the gaps in traditional workflow management - the process based approach. Case Handling bases process execution in what can be done in order to achieve the goal instead of executing a well defined sequence of steps. This idea is possible, by using the data that is produced as constraints that drive process flow.

The motivation for changing the way of thinking workflow was focused on the problems of the existent technology: Inability to lead with complex problems, change and exceptions. The way of work was too restrictive, thus inefficient with people bypassing the workflow constantly.

Comparing with classic workflow approach, the importance of information grows in case handling, reaching the level of the activities/sub-processes. Each instance of the process is handled separately. So, the process can be viewed as an instruction manual to handle the process information. There is no concept of activity ordering – only pre-conditions that must be satisfied in order to proceed the execution.

Two important entities on the Case Handling paradigm are data objects and roles. Data objects define the state of each case (process instance). Forms are entities used to see the state of the data objects in the case. Data objects can be connected to the whole process or with some particular activities, according to their type. They can be free (available for change in the whole process), mandatory (associated with a particular activity that cannot be concluded without its existence) or restricted (can be modified or created in defined activities). Case handling’s roles are different from traditional workflow in a way that coexist more than one per activity, in opposition to the only execution role. Known roles are execute (responsibility to execute the process or activity), redo (undo the executed activities), and skip (to bypass the activity). These entities, among others, are defined in a meta-model, described on literature [16].

2.3 Business Process Modeling

After studying the context in which the business processes will fit in the enterprise architecture and how they can be conceived is time to study the way how they can be represented. Different notations have been referenced.

2.3.1 Unified Modeling Language

Unified Modeling Language (UML) is a set of notations and diagrams that can be freely extended, by creating new profiles. The aim of this approach is to look concretely on the UML profile
for business process, included in UML 2.0. This profile is an adaptation from the system modeling notations to a business process context.

There are two distinct models in this UML profile [7]: Business Use case model, to describe the processes, as actions that provide some observable result for any of the business actors. The second model is the business object model, related with the way how a business process is executed by relating entities and executioners.

On business use case model one can find business use case diagram, in which business actors and business use cases are represented. Business actors are entities that interact with process – by initiating it or being interested in the outputs. UML activity diagrams can also be found here in order to perceive how the process can create value on the output.

The business object model is used to explain each process. The model is made of business class diagrams that relate the business entities one to each other like any other domain model. Activity diagrams, with more details than the ones from the first model, are used to draw activity routing. Sequence diagrams can also be used in this model.

On the evaluation of this notation, as positive characteristics there is the proximity with system modeling, that can be useful in the next stages; and the fact that not only the design and development team understand the diagrams. Management can understand it too.

The negative features on UML are the reasonable number of diagrams and the inability to represent other paradigms than the classic workflow (e.g. data flow).

2.3.2 Business Process Modeling Notation

Business Process Modeling Notation (BPMN) [17] is an Object Management Group (OMG) standard to business process modeling. The motivation for the existence of BPMN is the problem that UML produces a considerable number of diagrams.

The notation used in BPMN is heavily based on flow diagrams that have been adapted to represent business processes. A single diagram is provided (Business Process Diagram). An example is shown on Figure 3.

The used notation has a rich variety of symbols, such as activities, sequence flows, information flows, business events, processes, gateways, pools and lanes, data objects and annotations.

On evaluation of this notation, the positive aspects are agility, simplicity, direct mapping into execution languages and the existence of data objects, useful on data driven processes. As negative features BPMN has no support for different roles on the same activity, no chance to represent interaction and relationships between data objects.

2.3.3 Conclusion on Notations

The choice of which notation should be used is strongly influenced by the business area. Each business area may be directly connected with the main advantages of a notation in particular. Associations can be made between notations and business features. As an example if the business deals mainly with people and task delegations, UML would help to define business actors and hierarchies as well as explain the tasks as business use cases. BPMN would be helpful if the process would be implemented in a technology support by taking advantage of the direct mapping into execution languages.

Figure 3: Example BPMN diagram [19]

2.4 Business Process Execution Languages

The business process execution languages are as important as the notations. They represent a way to convert process definitions (from the business architecture) into functional technology elements (technology architecture). Based on past works [2], the aim is to analyze relevant languages that are used nowadays.

2.4.1 Business Process Execution Language

The Web Services Business Process Execution Language (WS-BPEL) [10] defined as a main goal the usage of globally accepted standards. The aim is to explore a platform that allows the integration of business interactions in common communication model. This language is supported by technologies such as web services and WSDL.

On evaluation of this language, the positive aspects are the standardization effort as well as the service behavior, like the ones that are invoked. On negative aspects, the complexity taken on normalization and specification details can be avoided in homogeneous systems.

2.4.2 jPDL

jPDL is a platform based business process execution language, which appeared in order to fight the complexity of WSBPEL, in homogeneous contexts. Despite of having a coded representation in XML, jPDL provides a graphical platform to define and observe the business processes in a flow diagram style. Almost all of the elements studied in the workflow and notations sections are available to use in this language.

Evaluating the language itself, its strengths are the simplicity (when compared to WSBEL), as well as the integration inside a single technology execution environment, the jBOSS jBPM, using only Java code. The ease of process development is granted in IDEs revealing the advantage of not writing any code. The lack integration with other technologies and platforms is a negative aspect, which is not relevant in the single platform philosophy.
2.5 Workflow Management Systems

This section addresses the place where the previously defined business processes will be executed. First the reference model for workflow management system, and then a concrete case of a workflow management system.

2.5.1 Workflow Reference Model

A Workflow management system will allow the task distribution among the participants in the business process. This distribution is done according to an order in a set of rules previously defined. Because of it, every Workflow Management System supports three functional components [9]:

- **Build-time functions** for defining the processes;
- **Run-time control functions** to instantiate and execute the process; and
- **Runtime Interactions** needed for external entities interaction on the process.

Based on these three components, the reference model on Figure 4 summarizes the typical architecture for a workflow management system.

2.5.2 jBOSS jBPM

jBPM is a workflow management system, based on J2EE (Java Enterprise Edition) platform, which is executed in J2EE compliant application servers. Currently, by default, it is used on jBOSS application server.

This workflow management system is the cause of existence of the already studied jPDL execution language. However more languages like BPEL or Pageflow are supported too, as well as user-defined languages.

On jBPM the process execution corresponds to navigation in the business process graph, using **tokens**. Each token defines an execution path, keeping a reference to the current graph node. The token receives signals that make it change state in the graph along the defined transitions. Each graph node defines the token’s behavior, between four possible: Do not propagate the execution (wait state); Propagate the execution (go along a transition); **Fork** (create a new token) and **Join** (terminate the tokens).

Evaluating this workflow management system, the positive features are: the extensibility of the process execution engine by the power of adding new process definition languages; the ease of adaptation to different Java based application containers and the graphical tools offered to define the processes. On the negative side there is the weak support for data driven processes.

The decision of choosing jBPM in strongly based on technology. A J2EE based environment, despite of which application server is being used. A reason for not choosing jBPM is the need to interoperate with other platforms, where some more generic workflow managers (based on WSBPEL support) would work better with the complexity of orchestration.

3. PROBLEM ANALYSIS

In order to achieve the main challenge of this work, the possibility of developing a process view over the Fénix System, a concrete process will be studied: Master’s Thesis process. Based on the existing regulations [8], the whole business process will be defined, as well as the environment – business actors, workers and domain model.

3.1 Master’s Thesis Process Definition

As a start point to the process definition, the research begun on the existing documentation – a set of documents published by the Executive Board [4, 8]. These documents show what are the activities needed to execute in order to proceed in the Master’s Thesis process. This information has been complemented with some knowledge from the actual participants in the process.

Two different moments in time are well split – **two sub-processes where defined**: the first, concerning with the Thesis’s proposal making, approval, publishing and application period; the second sub-process is related with document submission and evaluation. The beginning of the whole Master’s Thesis process is triggered by a previously defined deadline to deliver the Thesis Proposals to Grade Coordinator.

The business process diagram for the **first defined sub-process** can be found on the main text. The activities defined there are the following:

1.1. **Submit Thesis Proposal** (TP) (**Propor tema de dissertação (TD)**)  
1.2. **Approve TP** (**Aprovar TD**)  
1.3. **Review TP** (**Rever TD**)  
1.4. **Abandon TP** (**Abandonar TD**)  
1.5. **Publish TP** (**Publicar TD**)  
1.6. **Apply to Master’s Thesis Course** (**Inscrever à disciplina de Dissertação de Mestrado**)  
1.7. **Apply to TP** (** Candidatar a TD**)  
1.8. **Assign TP to Student** (**Atribuir TD a aluno**)  
1.9. **Confirm Assignment** (**Confirmar Atribuição**)  
1.10. **Cancel Application** (**Abandonar Candidatura**)  

This sub-process is wholly executed inside the Supervisor’s Educational Department.

The second defined sub-process diagram can also be found in the main text. The activities described on it are the following:

2.1. **Submit Master’s Thesis** (**Entregar Dissertação**)  
2.2. **Jury proposal** (**Propor Júri**)  
2.3. **Jury Revision** (**Rever Júri**)  
2.4. **Discussion date proposal** (**Propor data de discussão**)  
2.5. **Set Discussion Date** (**Marcar Discussão**)  
2.6. **Write Discussion Report** (**E escrever Acta**)  
2.7. **Submit Master’s Thesis Final Version** (**Submeter Versão Final**)
3.2 Business Actors and Workers

After process definition on the previous section it is the moment to define who executes the activities. There are two types of participants: Business Actors, those who are interested in the process and its output (external to the process), and Business Workers, that are internal to the process and react to business actors interactions, with no autonomous decision making power.

In the whole process, as business actors there are: Student (aluno); Supervisor (orientador); Grade Coordinator (coordenador de curso), President of Jury (presidente do Júri) and Scientific Board Member (membro do Conselho Científico).

The student is a Master’s Degree student, who has the prerequisites to apply to the Master’s Thesis course. The activities 1.6, 1.7, 1.10, 2.1 and 2.7 are assigned to this business actor. The supervisor could be either a teacher (docente) or an external expert (especialista de mérito) and is responsible for the activities 1.1, 1.3, 1.4, 1.8 and 2.8. The grade coordinator is a professor nominated by the Executive Board to assure the grade execution. Activities assigned to this actor are 1.2, 1.5, 1.9, 2.2, 2.3, 2.9, 2.10 and 2.12. The president of jury is one of the 3 to 5 Jury’s members, homologated by the Scientific Board. The activities assigned to this actor are 2.4, 2.6 and 2.11. The Scientific Board member is assigned with the non represented activities of Jury composition homologation and discussion report homologation. These activities are not visible for the process, which assumes them as black boxes.

The missing activities are executed by business workers such as the educational department, the library or the academic services members. These activities do not involve any kind of decision making.

3.3 Business Domain Model

The business domain model is extracted from the defined process, based on the existing data objects. Those objects have an internal state. Stateless entities are added to the model in order to link the entities from the process and to give meaning to the model. The defined model is presented in Figure 5.

3.3.1 Entities with Internal State

By the analysis of the defined process in the full text, data objects where defined along the activity flow. Those data objects have an internal state, which is modified in certain activities.

The entities with internal state defined on the business process domain are a full representation of those data objects. The defined entities are:

- **Thesis Proposal** (Tema de Dissertação) corresponds to the thesis proposal made by the [future] supervisor. The states of this information entity are: Proposed, Approved or Rejected, and after approved Published. The next states are Applied, Assigned and Confirmed. Before the assignment, the application can be canceled, going back to Published state. The Applied state can be avoided with direct assignment, as seen on the process.

![Figure 5: Master’s Thesis process domain model](image)

**Jury** (Proposta de Júri) corresponds to the Jury member constitution, which has to be homologated by the Scientific Board. The first state is Proposed. If the Jury is homologated, it goes to the state Homologated, and if not, goes to Non Homologated. After revision the Jury goes back to the state Proposed.

**Master’s Thesis** (Dissertação) is the document itself, plus other aggregated element such as the Extended Abstract and other sources. For this entity the states are: Proposed Version, when delivered for the first time, and Final Version when it is submitted after discussion.

**Discussion Report** (Acta) is the result of the thesis discussion. It is formally a document, with the following states: Proposed, without the evaluation. When evaluation is entered the state is Complete. After the homologation process, the state is Homologated, if the activity was completed successfully or Non Homologated if it was not completed successfully.

UML State diagrams are used to describe these state machines in full text.

3.3.2 Stateless Entities

A poor domain could be constructed using only the previously defined entities. In order to link them properly and to add expressiveness to the domain model, stateless entities were added to the domain, as seen on Figure 5.

**Teacher Record** (Registo Docente) is used to define the entity that is linked to the Master’s Thesis as a supervisor, to the grade as the coordinator, to a department as a member, as well as a member of the Jury.

**External Record** (Registo Externo) is used to identify external people involved as supervisors or Jury members on a Master’s Thesis.

**Graduated Record** (Graduado) is an abstract entity used to aggregate the relationships that can be performed either by a teacher record or by an external record.

**Student Record** (Registo Aluno) is used here to associate the author to his Master’s Thesis.
Department (Departamento) is used to aggregate the teacher records, as well as (shared or not) grades.

Scientific Area (Área Científica) is a division inside the department where all the same scientific area teacher records are. The Jury elements are chosen based on this criterion.

Grade (Curso) provides a place to put a student record, as well as a teacher record as coordinator.

4. FÉNIX SUPPORTED PROCESS
Since 2004, as basis for most administrative activities on Instituto Superior Técnico, there is an information system – Fénix.

4.1 Fénix System
Fénix System is an interesting case on software engineering applied on academic domain. It is a web based system where everyone who belongs to this institution can make administrative and bureaucratic tasks. The evolution of this system is a constant, gathering more and more functionalities over the years.

Technically, the software architecture behind the system is based on Martin Fowler’s three tiered architecture [5]. On the business tier four layers can be found: Presentation Logic Layer, Presentation Logic Layer, Service Layer, Domain Layer and Data Access Layer. A substantial part of the business logic is placed mainly in the Service Layer, making the Domain Layer thinner. Despite of being thin, the domain model is rich – near 240 classes exist.

The main problem on the conception is the lack of a well defined EA. A consequence of it is the total inexistence of business architecture, and no business processes associated with the implemented activities.

Research is being done at the moment in order to implement an EA perspective, using a SOA approach [6].

4.2 Domain Model
Focusing the approach on the concrete problem of study, the Master’s Thesis process, Fénix already have domain entities that can support the whole process. The problem is located in the process flow approach, where this work will fit.

Fénix domain model supports the Master’s Thesis business domain in two different parts of the system domain model.

The first part supports what was defined as the first sub-process in chapter 3 – thesis proposal and application to proposals. The second part is related with thesis execution and evaluation, like the chapter 3 second sub-process.

Figure 6 shows the portion of the domain model related with the first sub-process.

Similarities can be found while observing this domain and the one that was defined for the business process in analysis. The Thesis Proposal is called here Proposal, the Execution Degree, corresponds to the Degree, as well as the Student and the Student Record. This model was developed before Bologna Declaration application, so that the application to a Proposal can be made by group of students.

Figure 7 defines the system domain model part that supports the second sub-process.

As observed in the first part of the system domain model, the similarities between the two models continue to exist: The most obvious correspondences can be seen in the class Degree, Teacher (to Teacher Record), Thesis and Student (to Student Record). The class Thesis Evaluation Participants corresponds to the Jury in the business domain.

4.3 Master’s Thesis Process Support
On recent Fénix developments, mainly in the second part of the domain, classes with internal state can be found. The idea of building a process view in this concrete case has been there since the implementation. Based on those states, the system provides to the developer a global state. The global states will be used later on this work.

Examples of the states that can be accessed by the grade coordinator are the following:
A – Student applied to the Master’s Thesis course. No Jury defined
B – Jury Proposal on construction.
C – Jury proposal submitted for homologation.
D – Jury proposal rejected by the Scientific Board.
E – Jury proposal approved by the Scientific Board.
G – Documents submitted and confirmed by the supervisor.
H – Documents submitted, confirmed by the supervisor and evaluation inserted.
I – Thesis evaluated and approved by the Scientific Board.

Important notes are added to this state representation: This set of states is incomplete – it does not provide states for the whole process. An example of it is the case of a non-approved thesis by the Scientific Board.

**4.4 Conclusions**
In the analysis of the Fénix System, in the concrete case of the Master’s Thesis process, it can be observed that Fénix really provides support for the process domain and isolated activities. The problem is the activity orchestration support, in a business process perspective, which does not exist yet.

However, an effort has been observed in order to create the set of global states in the second part of the process, but it stills incomplete - because of that, it could not be used to define the solution presented on chapter 5.

**5. SOLUTION PROPOSAL**
The aim of this chapter is to go a step further from the set of the states previously defined by Fénix for the Master’s Thesis business process. So the work will start from the business process domain model, where the entities with internal state are. The objective is to define, based on those internal states, a global state for the entire Master’s Thesis business process.

**5.1 Scope of the Solution**
This solution has, as the main goal, the ability to observe the actual state of a desired Master’s Thesis. To do that a flow will be made, based on a state machine, which reflects a combined state of the internal states from process domain entities.

To achieve the goal, some essential requirements are stated:

- **R1.** Master’s Thesis view shaped as a business process;
- **R2.** Master’s Thesis processes listing according to what is relevant for the current user;
- **R3.** Presented data must be based exclusively on Fénix system, being the only trusted source of information;
- **R4.** The current state of the business process should be seen in a business process management system.

These requirements are essential to fulfill the initial challenge. The following requirements are desired to the solution, however they do not compromise the initial objective of this solution:

- **R5** Graphical view of the Master’s Thesis Process;
- **R6** Only relevant information should be observed, the states that are not relevant for the user activity should not be shown;
- **R7** Access control should be provided, to prevent users from seeing non relevant information (no regulated constraints).

**5.2 Solution Design**

**5.2.1 Global State Generation**
Using the previously defined entities with internal state, in problem definition (Thesis Proposal, Jury, Thesis and Discussion report), a set of states has been created. Based on the combination of the existing entities in a specific moment in time, global states have been achieved.

On the first part of the process the only active entity is the Thesis Proposal, so the global states reflect directly the state of this entity:

- A. Proposed Thesis Proposal
- B. Rejected Thesis Proposal
- C. Accepted Thesis Proposal
- D. Published Thesis Proposal
- E. Applied Thesis Proposal
- F. Attributed Thesis Proposal
- G. Confirmed Thesis Proposal

The second part of the process involves the three other entities, but in different moments: in the first moment, there is the Thesis and the Jury entities; on the last moment the Thesis coexists with the Discussion Report. The generated states are the following:

- H. Thesis submitted
- I. Proposed Jury
- J. Non Homologated Jury
- K. Homologated Jury
- L. Thesis Discussed
- M. Thesis’s Final Version Submitted
- N. Evaluation Inserted on Discussion Report
- O. Non Homologated Discussion Report
- P. Homologated Discussion Report

Note that in this second part of the process the global state names are not always related with the state names of the involved entities.

**5.2.2 Process Views**
In order to define the relevant information to each user (fulfilling requirement R6) the states are considered relevant for an actor because at least one of these reasons:

- The entrance in the state was caused by the actor;
- The actor is interested in the state output or in the next transition.

Base on these two reasons the states that are relevant to the actors of the process/users of the solution are stated on Table 2.

<table>
<thead>
<tr>
<th>Actors</th>
<th>Relevant States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>D E F G H K L M P</td>
</tr>
<tr>
<td>Supervisor</td>
<td>A B C D E F G H K L M P</td>
</tr>
<tr>
<td>Degree Coordinator</td>
<td>A B C D F G H I J K L M N O P</td>
</tr>
<tr>
<td>Scientific Board</td>
<td>I J K N O P</td>
</tr>
</tbody>
</table>

**5.2.3 Solution Architecture**
A proposed architecture for the solution is presented in Figure 8. This architecture in Fowler’s 3-Tiered Architecture, adapted to fulfill the need to communicate with the Business Process Management System (the process access component). The resulting architecture satisfies all functional requirements.

**6. SOLUTION IMPLEMENTATION**
Master Thesis Monitoring Tool, or MT2, is the result of a partial implementation of the proposed solution. This is a partial implementation because not all the requirements were implemented. Despite of it, the most important requirement of process view for the Master’s Thesis process has been achieved.
The requirement R1 was implemented in the solution – based on the states that should be given by Fénix, the system builds a process instance for each Thesis. Requirement R2 was fully implemented allowing each type of user (Teacher, Degree Coordinator, and Student) to list the previously defined correspondent theses. The Scientific Board should have access to all Theses – this can be done, while no authentication exists, by searching for a specific student.

Due to effort balance of this development, the requirement R3 was implemented using a Stub, in order to load a small amount of data to show process visualization and listing functionalities. Other limitations on technical resources level also caused the non-implementation of this requirement.

The requirement R4 is one of the strong points of this implementation. Each time the user observes a particular thesis it is possible to see how its instance is running on the workflow management system.

Non functional requirement R5 was accomplished. Since R6 and R7 were not essential to the main objective, they weren’t implemented.

6.1 Used Technology

This implementation of the solution was developed using Java Enterprise Edition (J2EE) platform. The result is a web application, running in a web application server, which is Tomcat 5.5. The coding step, as well as the process definitions were made using Eclipse IDE, with both J2EE and jBPM support.

As a business process manager, the choice was for a fully J2EE integrated workflow management system, JBoss jBPM. The chosen process execution language, jPDL, has a full graphical support on Eclipse IDE.

The integration with external systems, such as the jBPM system was done using JAR libraries manipulating the same data of the process manager. The integration with Fénix system was planned to be done using web services.

6.2 Application Architecture

The implemented architecture was exactly the same that was planned on the solution presented in chapter 5. No changes were made on the four tiered architecture.

6.3 Implemented Business Process

The implemented business process was defined using the states returned by the Fénix System. This option was taken in order to simplify the domain logic and consequently fulfill the process view requirement faster. This set of states was fully implemented using jPDL on the Eclipse environment, using graphical tools, and later deployed on jBoss jBPM system.

7. RESULTS

In order to see if the implemented application results fulfill the expectations, some test cases where created as well as a scenario to illustrate the MT2 usage.

7.1 Usage Scenario

To illustrate the MT2 interactions, the following scenario was created:

The Coordinator of the “Mestrado em Engenharia Informática e de Computadores” degree, wants to see if the student 53811 has already submitted the final version of his master’s thesis, and if his Supervisor already checked it.

Using the MT2 application in a web browser, the Coordinator inserts his number. After that a profile where all supervised theses and coordinated departments appears. By selecting the desired department, a list with all theses in execution appear. By selecting the right one, the coordinator is able to see the process state drawn in their screen (Figure 9 – see full text for more detailed screenshots). Following the jBPM link he can see now the process instance created in jBPM environment.

7.2 Test Cases

Some test cases proved that MT2 can support different specifications on the Master’s Thesis – theses with only one supervisor are allowed, as well as theses with one of the supervisors as an external person (External Expert). Changes on the thesis state were also tested.

7.3 Requirements Fulfillment

Full text contains a detailed table with all the requirement fulfillment analysis.

In general the implemented solution does not take all the results proposed on the requirements stated in the solution design (chapter 5). However there are some advantages that this implementation brings compared to the previous reality:

- Presenting Master’s Theses in a process view given by a state machine.
- Ease of access by the user to the information, by listing for each entity all available and relevant thesis processes.
• Connection to a process management system, in order to have a system that tracks the process orchestration and provides a new set of attributes to each process instance.

8. CONCLUSIONS AND FUTURE WORK

Remembering the main goal of this work - the possibility of having a parallel system to Fénix that allows having a process view built with data taken from the Fénix domain – a synthesis of every chapter’s main conclusions will be presented.

8.1 Executed Work

The first part of the work was concerned in the study of tools and methodologies available either to analyze the problem or to conceive the solution – chapter two showed that it is technically possible to build up a solution to this problem.

The chapter three was concerned in analyzing the problem – formalizing the business process was the first step, then defining the process actors and the business domain model. The main conclusion is the importance of having a good domain model to achieve the state variables.

Chapter four showed how the business domain model can be mapped in the system domain model. The conclusion is that it is not always possible to have a direct mapping of the business domain in the system domain model.

In chapter five the internal entity states were mapped into a global state. A reference architecture model for the solution was conceived. The conclusion taken of this chapter is the crucial impact of the state mapping in the initial challenge of building the parallel system. It is absolutely decisive before stepping into the implementation.

Chapters six and seven confirmed, with the implementation and a small requirements validation, the possibility of having a system that provides a process view for an existing system.

However not all cases are equal, and the main objective may not be achieved in more complex and less documented contexts.

8.2 Future Work

As a future work for this project two main directions can be followed: Formalizing the used methodology, that was almost empirically used here; and developing a generic technical solution, fully configurable to any other business processes. Both of the solutions reduce the risk of failure of this approach when leading with more complex problems. Other short term future work can be read on the full text.

9. REFERENCES


