ABSTRACT
The demand for transparency in business is growing. Several organizations and regulatory agencies are demanding more transparency. One approach that has been advocated by many is that businesses processes be documented and available for those wishing to obtain information on how the processes work. Notwithstanding, transparency is far from being achieved by just documenting processes and making them available, since there are several other characteristics pertaining to the transparency concern. However, transparency is an orthogonal characteristic to the predominant functional view of business processes. We explore this fact by understanding transparency as a cross-cutting concern with respect to business processes models. Traditional methods to business process modeling compound the several characteristics in a unified manner, which gives rise to complex models, where different concerns are tangled. Our work uses the case of transparency to show the need to apply concern modularization at the level of business modeling. We propose an aspect oriented approach to deal with the problem, and show how the proposal works for i*, when used as a business modeling language.

Categories and Subject Descriptors
[GENERAL]: D.2 SOFTWARE ENGINEERING D.2.1 Requirements/Specifications Languages

General Terms
Management, Documentation, Design, Languages

Keywords
Transparency, Aspects, Business Process Modeling

INTRODUCTION
Paraphrasing [1]: "A process is deemed transparent if it makes the information it deals with transparent (information transparency) and if it, itself, is transparent, that is it informs about itself, how it works, what it does and why (process transparency)."

In this context we define organizational transparency as: "the existence of organizational policies to give stakeholders information about access, use, presentation, understandability and auditability of organization characteristics" [2].

The demand for transparency has been in the agenda of social scholars as well as political leaders. In the middle of the 2008 Wall Street crisis, German Chancellor Merkel claimed for transparency from British and US financial institutions1. It seems that the earlier efforts of Senators Sarbanes and Oxley were insufficient [3].

Other initiative came from the necessity to solidify the financial systems in an international agreement and to fortify the financial stability worldwide: several national banking organizations established BASEL [4], a set of essential principles, created to contain the fragility of the banking system. In a different domain, the Extractive Industries Transparency Initiative (EITI) [5] aims to strengthen governance by improving transparency and accountability in the extractives sector.

Different authors provide different definitions for transparency. Holzner and Holzner [6] states that transparency is "The social value of open, public, and/or individual access to information held and disclosed by centers of authority". Lord [7] says: "Transparency is a condition in which information about the priorities, capabilities, and behavior of powerful organizations is widely available to the global public". Fung et al [8] uses the concept of target transparency: "Instead of aiming to generally improve public deliberation and officials’ accountability, target transparency aims to reduce specific risks or performance problems through selective disclosure by corporations and other organizations. The ingeniousness of target transparency lies in its mobilization of individual choice, market forces, and participatory democracy through relatively light-handed government action".

For the sake of exemplification: suppose a citizen may be willing to buy an artifact with some level of radiation, as informed by the producer, but may not be willing to buy that product, if people in the assembly line were exposed to higher levels of radiation. However to know if the latter case did occur, it would be necessary to know how the artifact is assembled. In this case you will need information on the process used to assemble the artifact: you will require that this process be transparent. The question is not why someone would need to know how the artifact is assembled, but the fact that there is a growing pressure from society towards the disclosure of process information, not only to fulfill curiosity, but mainly to support decision making.

Given this context, we may say that transparency is a quality characteristic referring to the level of disclosure of how a given

1 http://www.thelocal.de/14425/20080920/
process is enacted, revealing what it does, when, why and how it does it. So it is important to define how the organizations will model processes complying with transparency. This is a challenge. This paper reports an approach to tackle this problem.

1. PROCESS MODELING

To offer products and services, an organization has to follow processes. Such processes have objectives, policies, information (and as such, deal with documents), resources and are executed by roles represented by organizational agents. BPM (Business Process Management) is an area that studies organization processes modeling. Process modeling uses a set of concepts, models and techniques with the objective to develop and present the organization business model containing the elements cited above. The model results from an organization abstraction, considering its essential characteristics, from the business point of view. A model is a simplified vision of a complex reality. It is a graphical abstraction that represents the reality considering its prominent characteristics. The choice of the characteristics that will be adopted and the ones that will be discarded depends on the organization objectives. Thus, a business model is an abstraction of the reality and represents, in a simplified form, the elements of the organization [9].

A process model is a set of views that represent different perspectives of one or more specific aspects of the business. Together, these views allow for an agreement about the organization and its business serving as a base of communication, discussions about improvement and innovation. The business model is composed for a set of models, grouped to answer the critical questions about the organization business [9].

In general, the objective of a business model is to answer the following questions about the organization: What is being made? Who makes it? When is it made? Where is it made? Why is it being made? How is it being made? Answers to these questions require that we deal with the following concepts:

- **Goals**: they represent the objectives to be reached by actors in the processes,
- **Activities**: tasks needed to generate a specific product or service,
- **Resources**: represents the elements manipulated in the processes as, for example: documents, information and products
- **Actors**: all kind of stakeholders in the process

As organizations aim to achieve transparency, we believe that an effective strategy is to make their processes transparent. As such, we are imposing an extra quality for business processes models: transparency.

Understanding transparency as a required quality, leads to an important insight: transparency could be regarded as a non-functional requirement [10], and as such inherit important results in the area of requirements engineering [11]. As such, our proposal is to regard transparency as a softgoal.

2. TRANSPARENCY AS AN ASPECT

Understanding transparency as a softgoal is relying on the insightful observation about wicked problems made by Simon [12]. Simon coined the term satisfice as to denote situations where the solutions are not perfect but are accepted within some degree of relativity. Softgoals are different from goals or hard goals, because the latter is satisfied and the former are satisficed. Softgoals are related to wicked problems, since the social context will determine to which extent the softgoal has been met. Quality characteristics in the Non-Functional Requirements Framework [11] are modeled as softgoals.

We have reported in [10] our first attempt to encode transparency as a network of non-functional requirements using the NFR framework [11]. The NFR framework uses the concept of soft-goals, which allows a range of satisfaction related to the degree of how the “goal” is achieved. Figure 1 shows the transparency network.

Having transparency or not having transparency will not impact what the process does. The characteristic is general, and as such, spreads to different parts of a given process. As a softgoal, transparency can be seen as cross-cutting concern which can be a scattered and tangled in the process model. So, to introduce transparency in an organizational process model we are proposing an aspect-oriented approach.

![Figure 1 - Transparency Network from [1]](image-url)
3. AN ASPECT-ORIENTED STRATEGY TO PROCESS MODELING

As previously mentioned, as transparency is a orthogonal feature of process model, issues of scattering and tangling affects modeling, traceability and visualization of processes. This issue was addressed by Silva [13] who proposed a strategy for dealing with cross-cutting concerns at the requirements level. Since Silva defined a Meta model for dealing with cross-cutting concerns, independent of which language will be target of the aspectual defined a Meta model for dealing with cross-cutting concerns, with cross-cutting concerns at the requirements level. Since Silva was addressed by Silva [13] who proposed a strategy for dealing cross-cutting concerns integration Meta-model.

3.1 Meta model

The Meta-model for cross-cutting integration concerns is a strategy for process modeling that provides elements to register and analyze the cross-cutting interaction into process models. The strategy offers a way of thinking and shape cross-cutting concerns into process models in addition to providing mechanisms that facilitate the visualization, analysis and modification of models handled. The strategy consists of three activities called Identify, Insert and Visualize. In Figure 2 we can see the cross-cutting concerns integration Meta-model.

To Identify we provide a language and guidelines to define which cross-cutting concerns can be inserted in the process model. Together with these guidelines we propose the use of a cross-cutting concern catalog [14]. This catalog contains the cross-cutting concerns and its operationalizations. The operationalization is presented through a specification that defines which features will affect the processes. As such, choosing the operationalizations, in the catalog, we can add to the process model the elements that will implement the cross-cutting concern.

The Insertion is responsible for integrate the operationalizations in the model. This integration is carried out by the interpretation contained in cross-cutting relationships and application of rules of insertion. The rules of insertion define how the elements of the cross-cutting relationship are inserted.

The resulting integrated model allows that different views be handled by the Visualize activity. This activity provides some partial models, so as to facilitate the understanding about process characteristics.

Using an analogy with Silva’s approach [13], we have:

- The activity of insertion is equivalent to composition, differing only in implementation. The insertion activity is a non-automated activity and the addition is made through the insertion of new elements (operationlizations defined in the catalog) in the model.
- The engine to visualize has the same concept as proposed by Silva, differing only in some rules.

Figure 3 represents the analogy between the two approaches.

3.2 Guidelines for the use of strategy

It is important, before decide to use our approach of modeling cross-cutting concerns in processes, to verify if the characteristic that will be inserted in the process model are really a cross-cutting concern. Below there is a general check-list to help this decision:

- The activity of insertion is equivalent to composition, differing only in implementation. The insertion activity is a non-automated activity and the addition is made through the insertion of new elements (operationlizations defined in the catalog) in the model.
- The engine to visualize has the same concept as proposed by Silva, differing only in some rules.

Figure 3 represents the analogy between the two approaches.

To Identify we provide a language and guidelines to define which cross-cutting concerns can be inserted in the process model. Together with these guidelines we propose the use of a cross-cutting concern catalog [14]. This catalog contains the cross-cutting concerns and its operationalizations. The operationalization is presented through a specification that defines which features will affect the processes. As such, choosing the operationalizations, in the catalog, we can add to the process model the elements that will implement the cross-cutting concern.

The Insertion is responsible for integrate the operationalizations in the model. This integration is carried out by the interpretation contained in cross-cutting relationships and application of rules of insertion. The rules of insertion define how the elements of the cross-cutting relationship are inserted.

The resulting integrated model allows that different views be handled by the Visualize activity. This activity provides some partial models, so as to facilitate the understanding about process characteristics.

Using an analogy with Silva’s approach [13], we have:

- The activity of identification is equivalent to the separation provided by guidelines and a language, plus only a catalog of cross-cutting concerns.
Figure 3 – Analogy between the strategies to integrate cross-cutting concerns in requirements and process modeling

- The characteristic will be repeated many times in the model.
- If the characteristic being present or not in the process model does not preclude the realization of the overall objective of the process.
- The characteristic can be reused in other domains.
- The characteristic may be associated with many variables.
- The characteristic is independent, representing a concept.
- The characteristic can be arranged in taxonomy itself (catalog).
- The characteristic can be operationalized through actions applied to the elements of the model process language (goals, actors, activities, resources and others)

3.3 Strategy Components

Silva’s strategy defines: a language, guidelines on how to use it, a mechanism of integration and a mechanism for viewing. Similarly, our approach defines a language for Aspect Oriented Process modeling Meta Language (AOPML), guidelines on how to use it, mechanisms for insertion of cross-cutting concerns using an aspect catalog base and a viewing mechanism, Figure 4. The mechanisms used for entering and processing rules depend on the constructs of AOPML. Thus the identification, insertion and visualization are focused on languages for modeling business processes.

The infrastructure needed for the identification activity is composed by AOPML, Aspects Catalog and Guidelines for AOPML use. AOPML is composed by model components of any language for modeling processes, such as BPMN (activities, flows, gateways, events and others) [15] or i-star (i*) [16] (actors, goals, softgoals, tasks, resources); a core model which describes the cross-cutting relationships, and a joinpoints model that defines how the core model and components model are related and which elements of the components model may be involved in a relationship of the type cross. The Catalog is composed by cross-cutting concerns and its operationalizations. Guidelines for AOPML uses are composed by the rules to identify the operationalizations to be inserted in the process model.

The infrastructure needed for insertion activity is composed by mechanisms of insertion and rules of insertion that establish how the cross-cutting concerns will be inserted.

The infrastructure needed for visualization activity is composed by mechanisms of visualization and rules of visualization that establish how the cross-cutting concerns will be inserted.

3.3.1 Identification activity

The identification activity is composed by AOPML, Cross-cutting concerns Catalog and Guidelines for AOPML. Each one is detailed above.

3.3.1.1 Aspect oriented Process Modeling Language - AOPML

Structurally, the aspect oriented process modeling language (AOPML) consists of the following parts:

a) Components Model - defines the elements of the process modeling language.

b) Core Model - describes the cross-cutting relationships. It is a new kind of relationship to be added to the model of components;
c) Joinpoints Model - define how the core model and components model are related and which elements of the components model may be involved in a relationship cross.

Figures 5 and 6 illustrate the details of these models. The elements represented in white are the hot-spots (they can be instantiated); the others represent the fixed parts of the framework.

Figure 5 – AOPML syntax (adapted from [13])

Figure 6 – Oriented Aspects Process Modeling Language conceptual model – AOPML (adapted from [13])

Figure 6 – Cross-cutting relationship syntax from [13]
3.3.1.1 Core Model – Cross-cutting Relationship

The cross-cutting relationship has the label "cross" and inherits the details about origin and destination (pointcuts) of the relationship. In addition to this information about cross-cutting the relationship is described through pointcuts, advices and intertype declarations. The syntax of the cross-cutting relationship is illustrated in Figure 6.

3.3.1.1.1 Pointcuts

The pointcuts indicate the model elements affected by a particular characteristic. They are defined by a name and an expression that are sentences that use operands, operators and primitives. The operands are elements whose types are associated with joinpoints. Operators are used to group one or more joinpoints. They are: OR, AND, NOT. The primitives define the actions to be undertaken in the pointcuts. They are: include - adds a set of elements to the specified location, substitute - replace elements indicated or exclude - remove elements when the advice or intertype declaration has his body empty.

3.3.1.1.2 Advice

The advice defines which catalog operationalizations will be inserted in the process model. Each advice is formed by a type (AFTER, BEFORE, AROUND) that indicates how they will affect the pointcuts; an expression of pointcuts, indicating which are the pointcuts where they will be implemented, and the body that sets the group of elements that scattering and tangling in pointcuts. The body of each advice identifies the operationalizations to be included or replaced.

3.3.1.1.3 Intertype Declaration

The intertype declaration is used to add new types of elements into the process model chosen, indicating that these are new elements arising from the junction of two characteristics.

3.3.1.2 Guidelines for the use of the language

The use of aspect-oriented language to modeling cross-cutting concerns in process models needs to follow some steps as related above:
- Identify the components of the language (components model). In general they will be the language constructs;
- Identify the elements that can take part of joinpoints model, defining pointcuts, advices and intertypedeclarations. These elements will be affected by the inclusion of cross-cutting concerns;
- Identify cross-cutting relationship. Define the kinds of cross-cutting relationship that will be establish in the model;

3.3.1.3 Mechanism to identify cross-cutting concerns to be inserted in process models

The identification of aspects in process models is responsible for checking which aspects in catalogs will be inserted in a process model. It is a manual activity. The process manager has to decide which cross-cutting concerns should be implemented in the process. It is supported by the following guidelines:
- Receive as input the softgoals of the organization (characteristics);
- Identify using the Cross-cutting concerns catalog, the cross-cutting concerns that should be scattering and tangling in the process model to accomplish to the softgoal;
- Choose the operationalization to be inserted. Each cross-cutting concern may have more than one kind of operationalization;
- Analyze the contributions among cross-cutting concerns (make (++), help (+), unknown(?), hurt(-), break(--)) of each operationalization to avoid conflict;
- Generates as output a possible list of operationalizations to be included in the process model;

3.3.2 Insertion activity

The insertion activity is composed by mechanisms of insertion and rules of insertion that establish how the cross-cutting concerns will be inserted. It is done through the insertion of a cross-cutting relationship. The use of cross-cutting relationship reduces the numbers of relationships among model elements because represents a lot of interactions in cross-cutting relationships. This representation is possible because most of time a group of relationships is repeated to associate one characteristic to a lot of elements. The parts of insertion activities are detailed above.

3.3.2.1 Mechanisms an rules to insert cross-cutting concerns in process models

The integration of cross-cutting concerns in process models is responsible for scattering and tangling the operationalizations obtained from catalogs in the process activities. It must be guided by the semantic rules of the model and is responsible for scattering and tangling the operationalizations selected by the mechanisms of identification. It is supported on the following guidelines:
- Receives as input the process model in AOPML and the list of operationalizations generated in the identify activity;
- Inserts in the process model the activities and resources (operacionalizations) to promote the achievement of the cross-cutting concerns desired using the cross-cutting relationships. All the cross-cutting concerns are operationalize through activities and resources;
- Generates as output the process model with the operationalization of cross-cutting concerns. The activities and resources takes part of the process model;
- Analyze the contributions (make (++), help (+), unknown(?), hurt(-), break(--)) of each operationalization inserted in the process model. It can conflict with another one.

3.3.3 Visualization activity

The visualization activity is composed by mechanisms of visualization and rules of visualization that establish how the cross-cutting concerns will be visualized.

3.3.3.1 Core Model – Cross-cutting Relationship

The cross-cutting relationship has the label "cross" and inherits the details about origin and destination (pointcuts) of the relationship. In addition to this information about cross-cutting the relationship is described through pointcuts, advices and intertype declarations. The syntax of the cross-cutting relationship is illustrated in Figure 6.

3.3.3.1.1 Pointcuts

The pointcuts indicate the model elements affected by a particular characteristic. They are defined by a name and an expression that are sentences that use operands, operators and primitives. The operands are elements whose types are associated with joinpoints. Operators are used to group one or more joinpoints. They are: OR, AND, NOT. The primitives define the actions to be undertaken in the pointcuts. They are: include - adds a set of elements to the specified location, substitute - replace elements indicated or exclude - remove elements when the advice or intertype declaration has his body empty.

3.3.3.1.2 Advice

The advice defines which catalog operationalizations will be inserted in the process model. Each advice is formed by a type (AFTER, BEFORE, AROUND) that indicates how they will affect the pointcuts; an expression of pointcuts, indicating which are the pointcuts where they will be implemented, and the body that sets the group of elements that scattering and tangling in pointcuts. The body of each advice identifies the operationalizations to be included or replaced.

3.3.3.1.3 Intertype Declaration

The intertype declaration is used to add new types of elements into the process model chosen, indicating that these are new elements arising from the junction of two characteristics.

3.3.3.2 Guidelines for the use of the language

The use of aspect-oriented language to modeling cross-cutting concerns in process models needs to follow some steps as related above:
- Identify the components of the language (components model). In general they will be the language constructs;
- Identify the elements that can take part of joinpoints model, defining pointcuts, advices and intertypedeclarations. These elements will be affected by the inclusion of cross-cutting concerns;
- Identify cross-cutting relationship. Define the kinds of cross-cutting relationship that will be establish in the model;

3.3.3.3 Mechanism to identify cross-cutting concerns to be inserted in process models

The identification of aspects in process models is responsible for checking which aspects in catalogs will be inserted in a process model. It is a manual activity. The process manager has to decide which cross-cutting concerns should be implemented in the process. It is supported by the following guidelines:
- Receive as input the softgoals of the organization (characteristics);
- Identify using the Cross-cutting concerns catalog, the cross-cutting concerns that should be scattering and tangling in the process model to accomplish to the softgoal;
- Choose the operationalization to be inserted. Each cross-cutting concern may have more than one kind of operationalization;
- Analyze the contributions among cross-cutting concerns (make (++), help (+), unknown(?), hurt(-), break(--)) of each operationalization to avoid conflict;
- Generates as output a possible list of operationalizations to be included in the process model;

3.3.3.4 Insertion activity

The insertion activity is composed by mechanisms of insertion and rules of insertion that establish how the cross-cutting concerns will be inserted. It is done through the insertion of a cross-cutting relationship. The use of cross-cutting relationship reduces the numbers of relationships among model elements because represents a lot of interactions in cross-cutting relationships. This representation is possible because most of time a group of relationships is repeated to associate one characteristic to a lot of elements. The parts of insertion activities are detailed above.

3.3.3.4.1 Mechanisms an rules to insert cross-cutting concerns in process models

The integration of cross-cutting concerns in process models is responsible for scattering and tangling the operationalizations obtained from catalogs in the process activities. It must be guided by the semantic rules of the model and is responsible for scattering and tangling the operationalizations selected by the mechanisms of identification. It is supported on the following guidelines:
- Receives as input the process model in AOPML and the list of operationalizations generated in the identify activity;
- Inserts in the process model the activities and resources (operacionalizations) to promote the achievement of the cross-cutting concerns desired using the cross-cutting relationships. All the cross-cutting concerns are operationalize through activities and resources;
- Generates as output the process model with the operationalization of cross-cutting concerns. The activities and resources takes part of the process model;
- Analyze the contributions (make (++), help (+), unknown(?), hurt(-), break(--)) of each operationalization inserted in the process model. It can conflict with another one.

3.3.3.5 Visualization activity

The visualization activity is composed by mechanisms of visualization and rules of visualization that establish how the cross-cutting concerns will be visualized.
3.3.3.1 Mechanisms an rules to visualize cross-cutting concerns in process models

The visualization of cross-cutting concerns is responsible for generating visions of the process model so that the organization may have in these models only the elements necessary to their analysis. Often there are different goals for modeling a process. For example, a model built to establish the manner of operation of an organization may not necessarily need as many controls and audits as a financial process. Characteristics of various dimensions when modeled together can make the model incomprehensible or difficult to use. It must be guided by the semantic rules of the model. The mechanisms of visualization help to reduce the scope of a problem and thus its complexity. It is supported by the following guidelines:

- Receives as input the process model with all cross-cutting concerns modeled;
- Receives as input the cross-cutting concerns and cross-cutting relationship chooses to be viewed and the rules;
- Generates parts of the process model containing cross-cutting concerns and cross-cutting relationship selected.

4. THE USE OF THE STRATEGY

As reported in the previous section, our process modeling strategy is focused on the use of AOPML. Here, we will present an instantiation of AOPML for i* [16]. The following sections define respectively the identification, integration and visualization activities for this instance.

4.1 i* as the component model

“i* offers a framework for modeling, analyzing and designing business process in terms of intentional strategic actor relationships. The Strategic Dependency (SD model) describes a business organization in terms of the dependencies that actors have on each other in accomplishing their work. The Strategic Rationales (SR model) describes the reasoning that actors have about the different possible ways of organizing work. The framework can be used to assist participants and stakeholders in a business process to develop a deeper understanding about the existing process and to systematically generate alternatives in order to arrive at new process designs that can better address business objectives and individual concerns.

A business process is most often modeled as a network of flows of work products from one work unit (a department or a person) to another. A more detailed model would show activities performed within each unit with intermediate products as inputs and outputs of activities. Workflow models show the entities and activities involved in a work process but not the reasons for their existence or relatedness." [16].

As we are working with softgoals we understand that is very fundamental to work with a language which permits represent goals and softgoals, so we decided to use i*.

Considering modeling processes in i*, we described an instance of AOPML, called AOi*, and some guidelines for its use. Thus, AOi* is the i* model extended, allowing the descriptions of softgoals, goals, tasks, resources and actors, and the control of scattering and tangling problem. This extension adds the cross-cutting relationship in i* model, and thus, a new way of thinking about identification, insertion and representation of cross-cutting concerns.

4.1.1 The language

Figure 7 presents the elements and relationships of the AOi* model, an instance of AOPML elements. The components model is composed by softgoals, goals, task, resource and actor. The joinpoint model is composed by softgoals, goals, task and resource. The actor does not take part of joinpoint model because it cannot be operationalized. The relationship decomposition with the labels AND and OR and the relationship of contribution with the label MAKE(++) , HELP(+), UNKNOWN(?) , HURT(-) and BREAK(--) are represented through the aggregation component. So each component has an attribute called “decomposition label” that indicates which is the label of the relationship between it and the component father. The correlation relationship is defined through “relCorrelation”.

4.1.2 Joinpoints model

In the case of AOi*, joinpoints may be elements like softgoal, goals, task, resource, as can be seen in Figure 7. The instances of these elements can be referenced as pointcuts. They can be identified: directly by element name or identifier either directly by regular expressions indicating types and topics.

4.1.3 Core model cross-cutting relationship

The elements operand, advice_body and intertype_body are associated with joinpoints Figure 5. Here, we describe the semantic of these elements in AOi*instance.

4.1.3.1 Pointcuts

Pointcuts make reference to any kind of elements like softgoals, goals, tasks and resources. On AOi* pointcuts, the OR operator indicates that if the operands reference to the child of the same parent element so only one of the operands need to be changed, while the operator NOT exclude a particular operating among elements specified in pointcut.

4.1.3.2 Advice

Each advice defines a cross-cutting concern contained in the catalog that will impact the model in the given pointcuts. In the case of AOi*, we can define RNF (softgoals) and operationalizations for the advice. Each advice is one of three types before, after, around.

- Before – insert before (in parent element) of the pointcut as a decomposition of its parent element.
4.1.3.3 Intertype Declaration

Intertype declarations are used to modify the elements structure of components model. We have not explored the intertype declaration for the AOI* instantiation.

4.2 Transparency as an example

To illustrate our proposal we have used the transparency softgoal network shown in Figure 1. Each softgoal in this network can be detailed and defined by another network. In this paper we will focus on the auditability softgoal. It is composed by verifiability, validity, controllability, accountability and traceability. In particular we will focus on the controllability softgoal.

4.3 Identification of cross-cutting concerns

Our target process is the PhD/MsC program admission process of our Department. We posit that the organization is interested in providing more transparency as making sure that controllability is been taken care in the admission process. Looking to our strategy first we have to identify the cross-cutting concerns, and as such we follow the identification mechanisms and rules.

a) Receive Softgoal - Achieve goals controlled

- Around - insert elements as part of decompositions of affected elements, as if this element was necessary to reach the affected part.
- After - insert after the pointcut as a decomposition of its parent element

b) Identify the cross-cutting – Controllability / Achieve measured goals. Figure 8.

c) Choose the operacionalization - Calculate Estimate vs. Realized Figure 9.

d) Analyze the contributions – No conflict

e) List of operacionalizations – Activity “Calculate Estimate vs. Realized” and Resources “% Realized”, “% Estimated”

4.4 Inserting cross-cutting concerns

The second step is to insert the cross-cutting concerns into the process model. In the AOI* model we have two types of diagrams. One is the SD situation which represents the strategic dependencies among actors. The other is the SR construct which represents the resources and tasks used to achieve the goals. The controllability can be inserted in any part of the process. We suggest addressing it by a divide and conquer approach [17]: for each situation (relation between two actors) we analyze the necessity of controllability and apply if it is necessary. For this example we choose the relationship between PhD/MsC Program Committee and PhD / MsC Program Secretariat, as show in Figure 10. To insert we follow the insertion mechanisms and rules.

- Receives as input the process model in AOPML and the list of operacionalizations generated in the identify activity; - PhD/MsC program admission process SD and SR situations; Activity “Calculate Estimate vs. Realized” and Resources “% Realized”, “% Estimated”
- Inserts in the SD model the softgoal - see Figure 11
- Inserts in the SR model the crosscutting concern - see Figure 12. The cross-cutting concern can be seen in Figure 13.
Generates as output the process model with the operationalization of cross-cutting concerns. The activities and resources take part of the process model; see Figure 12.

Analyze the contributions (make (++), help (+), unknown(?), hurt(-), break(--)) of each operationalization inserted in the process model. It can conflict with another one. – No conflict

Figure 8 – Controllability Network

Figure 9 – Operationalization Specification

Figure 10 – Process part affected by Controllability
Figure 11 - Softgoal insertion in SD model

Figure 12– Cross-cutting relationship example
4.5 Visualizing cross-cutting concerns

The third step is to visualize the cross-cutting concerns. It is responsible for generating visions of the process model so that the organization may have in these models. In the SD situation, we can have different visions about process model. We can see only the cross-cutting, Figure 12, we can see the tasks and resources inserted in the model, Figure 14, or other kind of views. To generate visions we follow the visualization mechanisms and rules.

```
CROSS-CUTTING {
SOURCE= Controllability;
POINTERCUT Approve candidates: INCLUDE (calculate realized X estimated)
ADVICE AROUND: approve candidates {calculate realized X estimated}
POINTERCUT calculate realized X estimated: INCLUDE (% realized)
ADVICE BEFORE: calculate realized X estimated { % realized}
POINTERCUT calculate realized X estimated: INCLUDE (% estimate)
ADVICE BEFORE: calculate realized X estimated { % estimate}
POINTERCUT calculate realized X estimated: INCLUDE (approve candidates)
ADVICE BEFORE: calculate realized X estimated {approve candidates} }
```

- Receives as input the process model with all cross-cutting concerns modeled; - see Figure 12
- Receives as input the cross-cutting relationship description - see Figure 13
- Generates as exit parts of the process model containing cross-cutting concerns and cross-cutting relationship chooses. - see Figure 14
5. CONCLUSION

We explored transparency as an orthogonal characteristic that may tangle different parts of a business process. Our approach shows the possibility of applying concern modularization at the level of business modeling. We have also instantiated Silva’s Metal Model, thus providing another example of its applicability in the realm of early application of aspects. We provided a simple, but meaningful, example, as of how to reuse operationalizations from a concern repository. In our case we used the softgoal transparency as to demonstrate the proposal. In our work we assumed that the concerns are already available, and as such concern elicitation was previously done [18], [19], [20].

Our literature review found just one paper dealing with related issues, looking to processes models from the point view of aspects: Ramos et al. [17] present an approach for reducing complexity of i* using aspects. The basic idea is (i) to identify and modularize crosscutting concerns (model elements that affect several other elements in the same model), and (ii) to define composition rules for these crosscutting elements (or aspectual elements) that allow to recover the original model. They identify aspects in the existing i* models and replace them with a new construct, called “star”. This “star” contains the tasks removed from the model and some weaving rules to compose them into the model again. The tasks are removed and the “stars” are inserted as actor’s dependum. The article evaluates the new “stared” models with respect to understandability.

In our approach the insertion of cross-cutting relationships helps to reduce the complexity of the model too, but the context is different, consequently the steps, mechanisms and rules to do this are different. In this context we have a Business Process Model and we need to insert quality characteristics as to conform to new requirements. We identify these characteristics and obtain their operationalizations from a given catalogue. In our specific case, we are adding information into the model, but we, as Ramos et al., are using an aspectual framework. One important difference from [17] is that, in our case, the i* language remained the same. It is also important to point that can also be used for different business process languages. Future work will contemplate other languages and larger case studies.

6. REFERENCES