Incorporating Directives into Enterprise TO-BE Architecture

Sagar Sunkle, Deepali Kholkar, Hemant Rathod, and Vinay Kulkarni
Tata Research Development and Design Center,
Tata Consultancy Services,
54B, Hadapsar Industrial Estate,
Pune, India 411013
Email: sagar.sunkle,deepali.kholkar,hemant.rathod,vinay.vkulkarni@tcs.com

Abstract—To stay competitive, enterprises must respond to changes as effectively and efficiently as possible and ensure the employed courses of action, whether in response to change or even to optimize business as usual, fall within the purview of internal and external directives. Often, the traceability from change drivers that led to specific directives being applied to actual business rules implementing the directives is never captured in machine processable and analyzable manner, making compliance to directives hard to track and demonstrate. We present a model-based solution that enables a) modeling directives at various levels of detail on top of extended enterprise architecture-based models of enterprise, b) analyzing the models for compliance, and c) ensuring operationalization of directives. Initial explorations with a real world case study suggest that it might be possible to establish both top-down and bottom-up traceability for directives toward compliance checking.

Keywords: Directives; Policies; Rules; Enterprise Modeling; Intentional Modeling

I. Introduction

After developing 70+ large business applications in several domains, we have observed first hand that modern enterprises exist in a dynamic environment with change drivers like technology obsolescence and advance, competitive pressure, evolving market conditions, and regulatory compliance, resulting in changes across interconnected dimensions [1]. In order to stay competitive, enterprises need to respond to changes as effectively and efficiently as possible, while adhering to all applicable directives. This requires a coordinated treatment of what, how, and why of all of enterprise dimensions [1], [2]. Further, this treatment needs to include directives, since compliance is mandatory. Directives can be defined as guidelines that direct courses of action an enterprise can take. Enterprises need to function within the purview of external directives such as regulations from governing bodies in the domain/ geography of operation, as well as internal directives such as organizational policies targeted at making existing/planned operations efficient. Directives are usually formulated as abstract guidelines [3] and are left to individual business units to be interpreted and implemented based on local context. Interpretation is done by experts. However, demonstrating compliance to directives at a later point becomes tough, since traceability and history of decision making are usually not available. This is an important issue enterprises struggle with, since they are required to establish compliance with external and internal regulations.

An enterprise needs to interpret abstract directives a) by applying them to appropriate courses of action that it is employing (to make operationalization of business as usual (BAU) optimal with regards some criteria) or is planning to employ (to make operationalization of a planned course of action optimal in response to change) and b) by operationalizing them in the form of concrete rules. For their model-based treatment it is necessary to model concepts pertaining to courses of action, such as actors, actions, resources, and (soft) goals. Furthermore, it should be possible to compute alternate courses of action in which directives are to be applied. In this paper, we show how directives can be modeled in the enterprise change context and queried if requisite modeling and model processing machinery is available. The key components of our approach are- a) an extended enterprise metamodel that builds upon our earlier work [4] to enable modeling of directives at various levels of detail, b) a method for explicating, computing, and evaluating alternate courses of action for achievement of strategic goals and relating directives to specific courses of action, and c) a way to ensure operationalization of a given directive in the form of a business rule. Our specific contributions are twofold- first, we provide both visual and ontological modeling support for directives on top of models of what, how, and why aspects of enterprise, and second, we illustrate how ontology-based model processing machinery can be used to query directives in meaningful ways to demonstrate compliance through traceability.

The paper is arranged as follows. Section II motivates explicating and analyzing directives in the enterprise change context and outlines our approach. Section III shows using models of a real world case study how the extended enterprise metamodel enables computing alternative courses of action and operationalizing directives that apply to them. Section IV shows how meaningful queries can be constructed on these models to reveal adherence to specific directives. Section V discusses related and further work and Section VI concludes the paper.
II. MOTIVATION AND OUTLINE

Whereas enterprise architecture (EA) modeling languages enable holistic modeling and analysis of various enterprise dimensions, explicit design decisions help enterprise to explicate and evaluate alternative courses of action in response to change [4]. The change drivers could be external, originating in enterprise’s environment, such as regulations to be complied with, or they could be internal, such as a decision to initiate merger with an equal based on prevalent market conditions. When responding to either kind of change and also to improve upon its BAU state, enterprises embody policy decisions made at various levels in the form of directives. Directives help an enterprise in governing and guiding the selected course of action. Essentially, directives keep the enterprise on course and moving towards its strategic goals. The abstract concept of a directive is specified generally in terms of business policies at strategic level and implemented in terms of business rules at operational level. It is the business rules that are supposed to provide the leverage needed for building more effective and adaptable business solutions [3].

Business Motivation Model (BMM) describes these concepts and provides a metamodel for specification of these concepts. But, we believe that an actionable treatment of these concepts in the context of enterprise modeling requires extension to the metamodel and model processing machinery that can query and analyze directives as they apply to specific courses of action. To enable such a treatment, we build upon our earlier work [4]. The AS-IS architecture model coupled with modification and addition of elements and relations would indicate a specific TO-BE architecture for the chosen intentional alternative. Figure 1 shows the extended enterprise metamodel which integrates ArchiMate’s core metamodel with intentional modeling concepts.

The ArchiMate generic metamodel defines active structure entities (ASEs) as entities that are capable of performing behavior. These are assigned to behavior entities (BEs), which indicate units of activity. Passive structure elements (PSEs) are the objects on which behavior is performed [5]. Using the metamodel in Figure 1, we were able to articulate enterprise’s change context as follows- ‘enterprise ASEs use or create PSEs while performing BEs as means to ends that are goal(s) and/or soft goal(s).’ To be able to model and analyze enterprise directives, we extend this metamodel with relevant concepts from BMM, to create the enterprise’s integrated EA + intentional and motivational (IM) model as shown in Figure 2. Only the motivational concepts related to directives are shown in Figure 2 on top of existing concepts in metamodel from Figure 1.

This mapping enables us to model and analyze directives along the following lines- ‘enterprise ASEs are motivated by internal and/or external drivers to change. Assessments
of aforementioned drivers lead to both goals and directives that support the achievement of these goals. Enterprises use or create PSEs while performing BEs as means to ends that are (soft) goal(s). Enterprise uses directives in the form of business policies and rules to govern/ refine the courses of action comprising relevant ASEs, PSEs, and BEs that enable the enterprise to achieve desired (soft) goal(s).  

To enable such a treatment of directives, we need to be able to-

1) compute alternate courses of action given enterprise's EA+IM model.
2) model directives in terms of policies that should apply to specific course of action.
3) represent EA operationalization models for all valid intentional alternatives and extract the operationalization model of specific/desired alternative.
4) model business rule(s) to implement given business policy in the operationalization model.

With these models two kinds of queries can be readily asked-

1) Given a strategic goal, which directives apply to given specific course of action that achieves this goal? Also, which business process(es) in the operationalization models implement(s) the directive under consideration? Finally, which business rules do these business process use to implement the said directives?
2) Given a business rule, which directive does it realize? Assessment of which driver led to formulating the directive? And finally, a course of action for which strategic goal does this directive govern?

Using the first kind of query, one can get information about the origins of a business rule in a directive which was arrived at through assessment of some driver. If that driver is no longer relevant in current change context for instance, then this business rule need not be implemented any longer. Either way, in contrast to the common practice of using reference documents describing policies for demonstrating compliance, our approach elevates directives to entities that are modeled in a specific change context with traceability from strategic goals down to operational processes which achieve these goals and can be analyzed for compliance and communication.

In the next section, we first detail visual and ontological modeling support for directive related concepts on top of EA and IM models. We then elaborate the process of explicating, computing and evaluating alternate courses of action and directives using a real world case study.

III. Modeling Courses of Action and Directives

A. Visual and Ontological Modeling Support

We extended Eclipse Modeling Framework-based Archi by adding support for IM concepts. All the models illustrated in this paper are created with extended Archi. The visual models are imported into EA and IM ontology using Archi’s export mechanism by first exporting models to .csv files and then programmatically importing .csv file contents into the ontology.

We presented an ontological representation that captures ArchiMate’s core metamodel as well as layer specific metamodels in [1]. This representation was versatile enough for conducting change impact and landscape mapping analyses. For EA ontology modeling and analyses, reader is requested to refer to [1]. All intentional modeling relations namely, means-ends, task decomposition, contribution, and strategic

\[1\] Archi - http://archi.ectis.ac.uk/
dependency relations, are represented as *reified* relations [6]. For instance, a contribution link indicates not only which element contributes to a soft goal but also what that contribution is. The contribution link in intentional model is captured as an ontological class CTLink, instead of representing it as a relation. The source and target of contributions are captured via hasCTSSource and hasCTTarget object properties while the contribution itself is captured via hasCTValue. Means-ends links (MELink), task decomposition links (TDLink) and strategic dependency links (SDLink) are similarly represented. This ontological representation of intentional concepts is used to elaborate various modeling steps. The ultimate objective of these steps is, beginning with IM modeling of enterprise’s change context, to obtain TO-BE EA that represents a specific course of action refined with directives in response to some change.

**B. Capturing Courses of Action and Directives**

Integration tasks are a specialization of task concept in intentional models and represent *primitively workable* elements [7]. They are leaf elements in strategic rationale (SR) models of actors that do not depend on elements of SR models of other actors. An EA behavior entity always realizes an integration task, i.e., it operationalizes an integration task (sometimes using some additional EA elements). This set of EA operationalization elements should consist of a business process where a business policy is realized as a business rule. This is shown in Figure 2. In the ontological representation this translates to an MPolicy instance reflectedIn IntegrationITask instance. In the set of operationalization elements that realize this IntegrationITask instance, a BusinessProcess instance realizes MBusinessRule instance corresponding to the MPolicy instance. The map of MPolicy instance and MBusinessRule instances that realize this policy is maintained separately.

Since directives apply to courses of action, these need to be determined first. Intentional models help capture alternative courses of action available to enterprise for transforming from AS-IS to TO-BE EA and directives are applied to refine courses of action in doing so. For BAU situations similar modeling approach can be utilized. Our approach results in four steps as depicted in Figure 3 which shows various kinds of models and analyses supported at each step.

We use a case study of merger and acquisition (M&A) of two large wealth management banks to showcase how courses of action and directives are modeled. Our organization was involved during the M&A activity initiated by one of the enterprises. We refer to these enterprises as WM1 and WM2. Both enterprises were in retail brokerage with 10000+ Financial Advisors across 700+ locations in Country X. WM3 was to be formed with the expressed strategic goal of tripling WM1’s revenue and gross margin in 5 years with a strategic growth viewpoint where it needs to provide new and innovative products and services to its clients.

We use the models of products and services rationalization to elaborate various modeling steps. The ultimate objective of these steps is, beginning with IM modeling of enterprise’s change context, to obtain TO-BE EA that represents a specific course of action refined with directives in response to some change.

**Step 1a- Exploring Strategic Alternatives** The existence of the product and services problem is traced back to the key external driver which led WM1 to engage into M&A with WM2. This situation is depicted in Figure 4. Market condition at the time was interpreted by WM1 as ripe for M&A for extending WM1’s product portfolio with that of WM2. These are captured as an external and internal driver respectively. Furthermore, that the product portfolios of WM1 and WM2, which were more or less equal in size and reach, would be same and this would lead to many similar products became obvious.

An assessment of the InternalMDriver instance Similar Products is carried out which shows that products and services of the merged entity need to be rationalized. The strategic goal of revenue increment (IHardGoal instance Revenue to be Increased with Doubled Product Mix) must be satisfied when rationalizing the product mix that contains many products of WM1 and WM2 with common features. Chief Financial Officer (CFO) is the stakeholder with responsibility of overseeing the activities that will be constructed via means-end analysis of this goal.

Two choices are available to CFO, either prefer the product mix of WM1 which is the majority stakeholder of the M&A or rationalize the product mix as shown in the SR model of CFO. For the first choice, IntentionalITask instance Prefer the Majority Stake, task decomposition suggests that when preferring the majority stake, margin should be improved still, captured in ISoftGoal instance Margin be Improved_2. Means-ends decomposition of IHardGoal instance Majority Stake be Preferred indicates
that either all of WM1 products are retained while WM2’s are decommissioned or WM1 products are retained and enhanced with additional features from similar products of WM2. Similarly, Chief Information Officer (CIO) has three choices when operationalizing each product in the resulting product mix- either Integrate Product (applications) AS-IS, Enhance Product Applications, or De-integrate and Decommission Product Applications. Business Unit Head’s (BU Head) strategic rationale is modeled along similar lines. The dependencies between CFO, CIO, BU Heads are captured as required.

Upon completion of the IM modeling activity for a given problem, IntegrationITask instances can be assigned satisfaction levels. Assigning satisfaction values to IntegrationITask instances is tantamount to making selection in means to all the ends in all actors that are part of the IM model. For products and services rationalization problem, the six IntegrationITask instances shown in Table I indicate one possible complete course of action or strategy in resolving the products and services rationalization problem. We refer to this strategy as strategy R. By operationalizing these tasks corresponding goals will be achieved, leading to achievement of the root goal IHardGoal instance Revenue to Be Increased With Doubled Product Mix.

We have implemented the bottom-up label propagation algorithm in [8] to compute satisfaction level of the root goal. Presuming the aforementioned set of combined actor-specific alternatives satisfies the root goal, we need to be able to compute each actor’s routine, or set of tasks, for chosen alternative. This is shown in Figure 4 for CIO, where the routine to achieve Product Mix Be Operationalized with Regards IT is highlighted.

The ontological representation easily enables implementing the label propagation as well as computation of specific routines. A SPARQL\(^2\) query to get all the immediate tasks that are means to a specific IHardGoal instance is shown in Listing 1. A routine can be conceived as a graph with the root goal of the actor as its root and one or more IntegrationITask instances as leaf nodes. A routine is recursively built by starting at the root goal(s) of an actor and traversing till IntegrationITask instances are reached. From IntegrationITask instances, flow of control goes back to parent nodes which are instances of IntentionalITask type connected via reified TDLink instances. This is because task decomposition links have AND semantics, i.e., in order to perform a task, all of its decomposed elements are needed. Both actors’ routine computation and label propagation is carried out using queries similar to one shown in Listing 1 to traverse over means-ends, task decomposition, contribution, and strategic dependency links.

**Step 1b- Relating Directives to Courses of Action**

While a strategy is thus established, an assessment may also stipulate specific ways in which appropriate product features are to be arrived at, which result in creation of internal directives in the form of organization policies. For instance, both WM1 and WM2 used to offer features that would result in more balances from clients. Another assessment of the internal driver Similar Products targeted at retaining such features for the merged entity leads to a policy called Encourage More Deposits/Balances as shown in Figure 4. Further inspection reveals that this policy should be reflected in CIO’s IntegrationITask instance Enhance Product Application at an appropriate level of granularity.

Government regulations are external directives that are modeled as external drivers, as in BMM. Their assessment results in creation of a special kind of directive called regulation policy, tagged so for traceability, rest of the treatment being same as for other policies. E.g. Individual retirement account (IRA) regulations shown in Figure 4, assessment of which leads to a regulation policy that is reflected in CIO’s IntegrationITask instance Enhance Product Application, indicating that when operationalizing IRA product the requirements of this particular regulation need to be effected in appropriate setting.

**Step 2- Capturing Problem-specific AS-IS Details** From the problem-specific IM model, it becomes clear which actors are key to the specific problem context. The precise

---

**Listing 1: Traversing Reified Means-Ends Links**

```sql
1 "SELECT ?task + (" + "
2 "+ " ?s rdf:type :IHardGoal . " + " ?s :name " + 
3 "<IHardGoal> + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + " + 
```

---

**Table I: Strategy R- Implementable Means to Ends Leading to Root Goal Satisfaction**

<table>
<thead>
<tr>
<th>IntegrationITask Instance (as means)</th>
<th>(To an end that is a) IHardGoal Instance</th>
<th>(Carried out by) IActor/ MStakeholder Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Retain WM1 Products, Enhance with Features from WM2</td>
<td>Prefer the Majority Stake</td>
<td>CFO</td>
</tr>
<tr>
<td>2 Enhance Product Applications [of WM1 products with features from WM2 and integrate with non-product applications]</td>
<td>Product Mix be Operationalized with regards IT</td>
<td>CIO</td>
</tr>
<tr>
<td>6 Cross Sell Products</td>
<td>Product Mix be Operationalized with Regards Sales Financial Advisors</td>
<td>BU Heads</td>
</tr>
</tbody>
</table>

---

\(^2\)SPARQL- http://www.w3.org/TR/rdf-sparql-query/
process in which key actors for a problem interact becomes easier to pinpoint through IM models when modeling the AS-IS architecture in terms of existing business processes, services, and underlying application services and functions and other EA elements. Directives need to be percolated down to operationalization by applying them to relevant business processes that must be included in the EA elements.

**Step 3- Creating EA Elements for On-ground Operationalization** In this step, IntegrationTask instances are related to EA elements that will operationalize them from TO-BE EA perspective. It is possible that in some cases, some AS-IS elements could be reused. IM alternatives can be combined in various ways based on whether they satisfy the root goal. In order to preserve which elements were added for specific leaf tasks, we tag them in the ontological representation using an object property called isOperationalizationElementOf.

**Step 4- Extracting Desired TO-BE Operationalization Model** The TO-BE EA model that captures all possible TO-BE alternatives would consist of the IM model, the operationalization model of all alternatives and the AS-IS model. An extended Archi viewpoint is created for the operationalization models of all alternatives and exported as the sole model to import into ontological representation. Using this as the base model, the set of operationalization elements of a specific strategy can be easily separated out using a SPARQL query over the tagging introduced in Step 3.

For strategy R, EA operationalization elements computed by the procedure detailed above is shown in Figure 5. The TO-BE EA model essentially consists of operationalization models of all possible strategies including Strategy R on top.
of AS-IS EA model

C. Operationalizing Policies with Business Rules

At the time of operationalization, various elements in the operationalization model are represented at finer level of granularity as appropriate. In the operationalization model of Strategy R shown in Figure 5, IntegrationTask instance Enhance Product Applications is realized by two services, one for product application enhancement and another for integrating enhanced product application with the rest of the applications. Particularly, application enhancement service is used by application enhancement process. This BusinessProcess instance Product Application Enhancement Process is further detailed in the form of a business process diagram. This is shown in Figure 6.

At finer level of granularity, further interactions between IT department (to whom product application enhancement is delegated via other organizational roles) and BU heads and Financial Advisors (FA) are captured as shown in Figure 6. BU Head first selects existing features of individual products from WM2 set of products. Then overall product enhancement business requirements are specified and relayed to IT. IT does the needful and rolls out enhancement changes to FAs. BU Heads train FAs on new features. FAs take stock of impact on clients and accordingly use the enhanced product applications.

In Figure 4, the MPolicy instance Encourage More Deposits/Balances was reflected in IntegrationTask Enhance Product Applications. The policy needs to be effected on ground, in the operationalization element Product Application Enhancement Process, specifically, in the BU Head’s selection of product features so that more deposits/balances are indeed encouraged on ground. Multiple control cases could be arranged since a policy may be translated into number of cases. Towards this end, the policy is first interpreted in the context of product feature selection in the form of the following concrete business rules that the products should comply with:

- Maintain $1000 balance for Platinum Tier Benefits: To get the ‘Platinum’ tier/category benefits, a balance of $1000 needs to be maintained in the client account. Before opening account in this category, make sure the initial deposit is more than $1000.
- For Gold Tier account, fee will be waived for the first 8 MF transactions in a FY, to encourage Silver category clients to upgrade to Gold category.
- For Platinum Tier account, fee will be waived for the first 20 MF transactions in a FY, to encourage Silver/Gold category clients to upgrade to Platinum category.

To apply the rules, we use a method that enables modeling constraints associated with a business process [9]. We view business rules as manifestation of a business policy that constrains some part of the business process in which they are applied, acting as non-functional requirements on the business process. In [9], high-level non-functional requirements on business processes are captured using operating condition and control case concepts. The operating condition serves as a classification or grouping of constraints, hence it is a high-level view of potential non-functional requirements to be defined. The operating condition signifies an applicable constraint and is an associated artifact to a business activity. When an activity is identified as having some associated operating condition, it may be necessary to further define control mechanisms that are to be put in place to control the operating condition. An activity operates under an operating condition and the operating condition is controlled by control case.

Effecting one such rule for the MPolicy instance Encourage More Deposits/Balances is shown in Figure 6. The operating condition recognizes the MPolicy by the same name as more deposits/balance policy condition. The corresponding rule is treated as the control case, here it is Maintain $10000 balance for Platinum Tier Benefits. The actual rule description contains further details.

In Figure 4, we modeled government regulation regarding IRA. The concrete rules defined for implementing this policy are as follows:

- Maintain $1000 balance for Gold Tier Benefits: To get the ‘Gold’ tier/category benefits, a balance of $1000 needs to be maintained in the client account. Before opening account in this category, make sure the initial deposit is more than $1000.

This IRA regulation policy and the corresponding rules can be attached to the same BusinessProcess instance Product Application Enhancement Process in a similar manner.

Note that the fine grained model of a business process is stored by reference to the BusinessProcess instance Product Application Enhancement Process in the operationalization model shown in Figure 5. Separate maps of operating conditions and MPolicy instances and control cases
and MBRule instances are stored.

IV. ANALYZING DIRECTIVES

Earlier in Section II we stated that two kinds of queries can be readily made over the directives modeled in the way detailed in the previous section. Given a strategic goal, finding out which directives apply given specific courses of action to achieve this goal becomes straightforward with the ontological representation. Presuming that for the goal of increasing revenue with the doubled product mix as shown in Figure 4 alternative evaluation has already suggested Strategy R, one needs to query this strategy. A strategy is stored in terms of the IntegrationITask instances as illustrated in Table I. Therefore, first the list of IntegrationITask instances of Strategy R are retrieved and then for each instance, query shown in Listing 2 returns MPolicy instances that were related by reflectedIn relation to this integration task.

Listing 2: Retrieving Policies Reflected in a Strategy

```sparql
1 "SELECT ?Policy " + "[ " +
2 "?s rdf:type :IntegrationITask . " + "?s :name " +
3 "<IntegrationITask> + "] . " +
4 "?Policy :reflectedIn ?s . " +
5 "]" ;
```

Also, with regards question of given this directive, which business process(es) in the operationalization model implement it, first EA operationalization elements of the IntegrationITask instance obtained in by execution of Listing 2 can be retrieved as shown in Listing 3. Query in Listing 3 returns all the operationalization elements of an IntegrationITask instance.

Once the operationalization elements are obtained, every BusinessProcess instance in these can be queried for operating conditions and control cases via corresponding maps.

The bottom-up queries are about obtaining specific directive(s) that given business rules realize. Starting with a business rule, BusinessProcess instance in which it appears as a control case is obtained. The IntegrationITask instance of which this BusinessProcess instance is an operationalization element can be obtained with a slight modification to Listing 3. Similarly with slight modification to Listing 2, policies that were reflected in this IntegrationITask instance can be obtained. Once specific MPolicy instances are obtained, querying a particular assessment that led to this policy is straightforward.

Listing 3: Retrieving EA Operationalization Elements

```sparql
1 "SELECT ?intgTask ?operationalEA " + "[ " +
2 "?s rdf:type :IntegrationITask . " + "?intgTask :name "]" +
3 "<IntegrationITask> + "] . " +
4 "?intgTask rdf:type :IntegrationITask . " +
5 "?operationalEA :isOperationalizationElementOf ?intgTask . " +
6 "]" ;
```

For the second part of the bottom-up query, the specific root goal, achievement of which is governed by implementing given business rule is obtained by traversing actor-specific routines in reverse direction starting with IntegrationITask instances. These are the IntegrationITask instances, operationalization elements of which contain the BusinessProcess instance that implements the given business rule as a control case.
Summary of Modeling and Analyzing Directives

With visual modeling support and ontological representation, it becomes possible to model and analyze directives in the enterprise change context. Since directives govern courses of action, a way of modeling and computing courses of action is needed. We have elaborated steps to be used for this purpose in the previous section. By establishing top-down and bottom-up traceability, questions about directives and rules that implement the directives can be answered by juxtaposing them with the goals and operationalization models of those goals.

V. RELATED AND FURTHER WORK

Regulations as External Drivers

It is recognized in [10] that companies face constant changes in regulations, such as Sarbanes-Oxley, Basel II, and HIPAA and with globalization companies become accountable for increasingly complex reporting requirements and need to increase the likelihood that data needed for compliance is readily available or can easily be accumulated. It is asserted in [11] that adherence to a regulation should consider the entire organization lest it results in operational disruptions. We have provided a general treatment of directives including regulations such that aspects related to a directive are explicated in the enterprise change context and queried for both top-down and bottom-up traceability aiding in assurance of compliance.

The high level compliance process in [12] envisions 5 steps—establish compliance framework, assess issues including risk, identify and implement internal controls, communicate controls, monitor controls and implement changes if necessary. Our treatment of directives is not unlike this process where modeling and model processing machinery is used to establish controls (in terms of operating conditions and control cases) and are available for querying for any communication purpose. An attempt is made in [13] to formulate the EA compliance problem as the problem to make EA comply with requirements that relate to a law and provide assurance that an independent auditor will accept this as conclusive evidence. We have specifically considered business and IT perspectives whereas [13] suggests to consider legal and political perspectives as well. Legal and IT dimensions are considered in [14] such that for every law, an IT compliance artifact is created at process and application level as required. At appropriate levels of detailing our approach can also be suitably extended to handle such perspectives.

Automated Policy-Rule Compliance Checking

Existing work on automating compliance checking makes use of linear temporal logic (LTL)/computational tree logic (CTL) based notations, complexity of which prevents adoption by business users who are typically responsible for compliance [15]. Rule/policy specification formats are either directly LTL/CTL [16] or higher level languages/notations translated to these, viz. constraint languages such as OCL [17], visual notations such as object diagrams [18], and policy languages that use markup formats and ontologies representing domain knowledge [19], [20]. Rules and policies defined in OCL are used to annotate business process models and used for checking compliance of processes using the SAL model checker in [21]. Tasks and rules are specified as object diagrams to compute rules applicable to tasks and check for compliance using SAL in [18]. Kharbili et al. [22] define a domain specific modeling language and method for regulatory compliance specification and automated checking called CoREL, targeted at business users. A graphical interface makes specification easy, however, the underlying rules for checking need to be specified in LTL/CTL which is hard. These approaches do not model enterprises holistically as in our approach nor consider directives at the level of enterprise models. Consequently rules only apply to courses of action at application level. As such, traceability from and to strategic goals is missing in these approaches. A syntax for specifying business rules is given in [23] targeted at supplementing enterprise models such as EKD, however, functionality to process directives is not dealt with as in our approach.

In our current approach, policies are reflected in a specific IntegrationTask instance by the modeler in consultation with the domain expert. Pinpointing a specific task where the policy applies may not always be possible for the domain expert. Nevertheless, the domain expert might be able to say which stakeholders, which resources, or which activities could be affected by a policy from which it should be possible for the modeler to use our model processing machinery to deduce specific IntegrationTask instances where the policy can be reflected. Also our model processing machinery could be used so that given a directive and additional information by domain expert about stakeholders, resources, and activities, it becomes possible to provide the related operationalization elements as context to help the domain expert to actually derive business rules from top level directive.

VI. CONCLUSION

When responding to changes, enterprises need to comply with directives, both internal and external (regulations). Directives apply to courses of action that enterprise employs in response to change. It should be possible to compute top-down traceability (from directives and strategic goals down to rules) and bottom-up traceability (business processes back up to directives and strategic goals). This enables an enterprise to be able to establish that a certain directive is implemented as a rule or that a business process basically implements a directive which came about by assessment of a specific driver at strategic level. We have shown using a real world case study that the integrated metamodel consisting of EA, IM, and directives related concepts makes modeling and model processing of directives intuitive by bringing out their relation with the courses of action. Using this
mechanism, we hope to address the problem enterprises face in establishing traceability to demonstrate compliance.

**REFERENCES**


