Association for Information Systems
AIS Electronic Library (AISeL)

ACIS 2006 Proceedings | Australasian (ACIS)

1-1-2006

# Correlating Business Process and Organizational Models to Manage Change

George Koliadis
*University of Wollongong, Australia*

Aditya Ghose
*University of Wollongong, Australia*

Moshiur Bhuiyan
*University of Wollongong, Australia*

Follow this and additional works at: http://aisel.aisnet.org/acis2006

Recommended Citation
Koliadis, George; Ghose, Aditya; and Bhuiyan, Moshiur, "Correlating Business Process and Organizational Models to Manage Change" (2006). *ACIS 2006 Proceedings.* Paper 28.
http://aisel.aisnet.org/acis2006/28

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2006 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

# Correlating Business Process and Organizational Models to Manage Change

Mr. George Koliadis, Prof. Aditya Ghose, Mr. Moshiur Bhuiyan
Decision Systems Lab,
School of I.T. and Computer Science
University of Wollongong,
N.S.W. Australia

## Abstract

*Business Process Management (BPM) provides the methods, tools and modelling notations to support a process-centric organizational view and management capability. As organizations grow in size and complexity, process improvement initiatives may involve change that has direct / significant impact across an organization. Thus, we provide methods and extensions to existing process modelling notations to analyse change against high- level models of the organization. Our approach permits improved analysis against higher-level organizational structures, motivations, inter-dependencies and capabilities that should be ideally considered as primary requirements during process design. Additionally, the organizational model becomes the 'scaffolding' with which to construct effective process architectures and management portfolios. This paper discusses our approach in the context of two modelling notations – the i* framework as an organizational modelling notation, and the BPMN notation for business process modelling.*

**Keywords**

Business process management, business process modelling, organizational modelling, change management.

## INTRODUCTION

A *Business Process* is a set of *dynamically coordinated activities*, controlled by a number of *socially dependant participants*, aimed towards the *achievement of a specific operational objective* (Smith et. al. 2003; Hammer et. al. 1993). Business Process Management (BPM) aims to support the effective (i.e. and, where possible, automated - Smith et. al. 2003) management of business processes within an organization via specialized tools and methods. The underlying premise of BPM is that the explicit modelling of the processes in an organization provides the level of (shared) understanding required to support effective organizational management and improvement practices (Harmon, 2003).

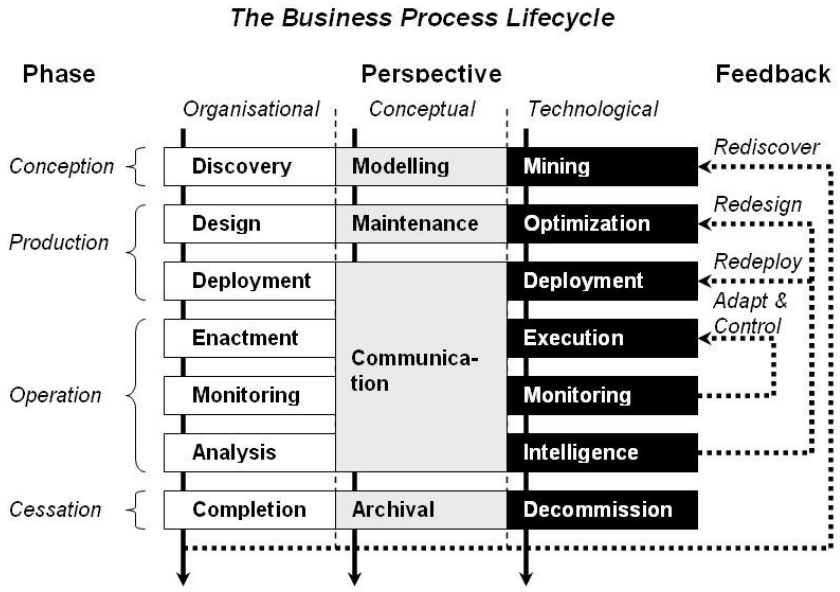



Figure 1: The 'Business Process Lifecycle' (BPLC) (Smith et. al. 2003).

Business processes evolve throughout their lifecycle (Smith et. al. 2003), triggered by change within their environment of operation. Figure 1 illustrates the major phases of the business process lifecycle. The lifecycle can be viewed from three primary perspectives - organizational, conceptual, and technical. In addition, four categories of feed-back and improvement mechanisms are represented – adapt & control, re-deployment,

redesign, and rediscover. Business process modelling aims to facilitate effective process change during the business process lifecycle via communication.

In Harmon (2003), three broad categories for business process change are defined – (1) *business process [re]design,* (2) *business process improvement*, and (3) *business process automation*. Both process redesign and improvement are a result of an analysis phase during BPM whereby process monitoring information is assessed against some performance criteria or benchmarks. Ad-hoc change on the other hand (as described in Aalst, 2003), refers to run-time adaptation, required given some unforeseen or isolated circumstance[s] (e.g. ignoring an insurance check requirement for an emergency patient). Ad-hoc change may occur prior to (i.e. pre-planned pre-enactment), or post initiation of a business process instance (i.e. on-the-fly at run-time).

Business process design should be conducted with reference to higher-level organizational motivations. Subsequently, our argument is that the development of business process models should reference principled high-level contextual models of the enterprise that illustrate its motivations, resources, social context, and internal/external inter-dependencies. Any purposeful changes made to business process models must be reflected within the high-level model for analysis against the greater context of the enterprise. To support an analyst in achieving this task, we offer methods to assess change between organizational and business process models. These methods support change management in the following sense: when changes occur to the high-level model these can be reflected in related business process models for eventual deployment, and when changes are proposed within a business process model these may be reflected at a higher level to improve analysis and decision making. In particular, we employ the use of an agent/network-oriented organizational modelling notation – *i** (Yu, 1995), and a standardized, operational and executable process modelling notation – BPMN (White, 2006).

We discuss the chosen notations further in the sub-sections below. We then describe our approach in detail, including our proposed process modelling extensions and analysis techniques. Finally, we illustrate our approach with examples, and conclude.

### Organizational Modelling (with *i**)

*i** (Yu, 1995) is an organizational modelling framework that supports a representation of the social, intentional, and strategic aspects of organizational structures. Specifically, *goal*, *soft-goal*, *task*, and *resource dependencies* can be modelled to help in understanding important strategic relationships between actors in an organizational context. From this perspective, their *motivations*, level of *commitment* and *vulnerability* can be effectively portrayed to support enhanced analysis and redesign capabilities.

Figure 2 represents a simple *i** Transport Organization model where (3) actors are represented in the context of 'Package Routing': a Sort Facility (SF); Bond Department (BD); and, Regulatory Agency (RA). In *i** actors are represented as circular nodes with links that illustrate their dependencies with other actors.

*i** provides two perspectives with which to view an organization: a Strategic Dependency (SD) model providing a high-level view of actors and their dependencies; and, a Strategic Rationale (SR) model illustrating each actors underlying motivations and capabilities. The SR model facilitates and understand of why an actor delegates, or is delegated, responsibilities in some organizational context.

When interpreting a dependency, the 'D' annotated to a link directs the dependency relationship from a *depender* (e.g. the 'Regulatory Agency') for a *dependum* (e.g. 'Bonded[Packages]') to a *dependee* (e.g. the 'Bond Department'). Each dependency may require either: a *goal* to be achieved (e.g. 'Bonded[Packages]'); a *soft-goal* to be satisficed (e.g. 'Timely Release[Packages]'); a *task* to be completed (e.g. 'Provide[Packages]'); or, a *resource* to be provided (e.g. 'Package Details'). An actors internal motivations and capabilities in an SR model, are represented as an AND/OR goal graph (as in Figure 2). Greater detail is available in an SR model concerning the source and destination task of dependencies between actors.

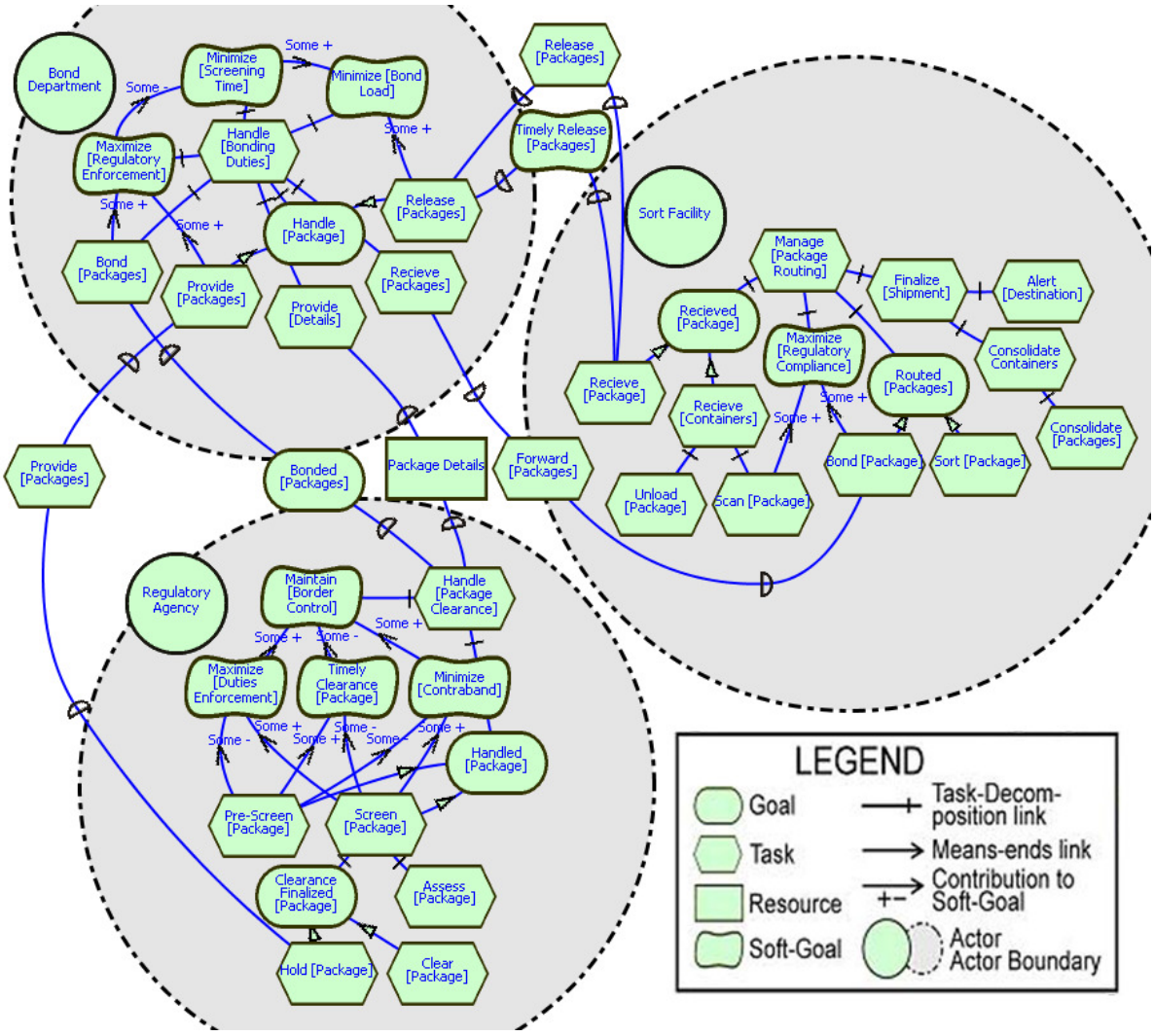



Figure 2: A simple *i** model of a Transport Organization.

**Business Process Modelling (with BPMN)**

The Business Process Modelling Notation (BPMN), developed by the Business Process Management Initiative (BPMI.org) (White, 2006) is primarily a technically-oriented business process modelling notation that supports the assignment of activity execution control to entities within an organization via 'swim-lanes'. BPMN has the capability to map directly to executable process languages including XPDL (Fischer, 2005) and BPEL (White, 2006; Ouyang, 2006). Furthermore, an analysis of BPMN (Becker et. al, 2005) also stated its high maturity in representing concepts required for modelling business process, apart from some limitations in terms of representing state, and the possible ambiguity of the swim-lane concept.

Figure 3 represents a Package Routing process in BPMN. Processes are represented in BPMN using **flow nodes**: *events* (circles), *activities* (rounded boxes), and *decisions* (diamonds); **connecting objects**: *control flow links* (unbroken directed lines), and *message flow links* (broken directed lines); and **swim-lanes**: *pools* (high-level rectangular container), and *lanes* partitioning pools. These concepts are further discussed in White (2006).

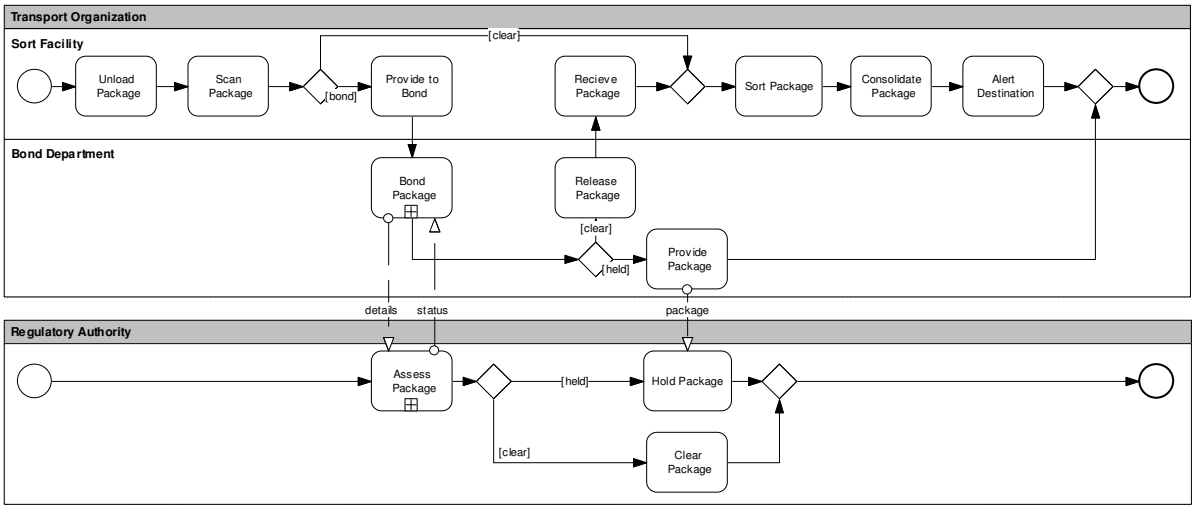



Figure 3: A BPMN 'Package Routing' process model.

# USING MODEL ANNOTATIONS TO ASSESS RELATIONSHIPS BETWEEN BUSINESS PROCESS AND ORGANIZATIONAL MODELS

*Activities* and *Sub-Processes* (i.e. represented in BPMN as rounded boxes), and *Tasks* (i.e. represented in an *i** model) signify a number of possible state transitions. The labelling of an activity (e.g. 'Register New Customer') generalizes one or more normal/abnormal outcomes (e.g. 'A new customer registered', or 'An attempted registration has been refused'). As such, most process and organizational models do not adequately represent enough information for effective analysis. They are too 'high-level', and do not convey a usable understanding of achievable states. In order to improve the description and clarity in process models, we propose to augment state altering nodes (i.e. activities and sub-processes) with semantic *effect annotations*. This parsimonious extension to the BPMN notation permits modellers to annotate activities in a process model with richer specification of immediate effects.

## Effect Annotation

An *effect* is the result (i.e. product or outcome) of an activity being executed by some cause or agent. Effects are commonly referred to as 'post-conditions'. An *effect annotation* relates a specific result or outcome to an activity on a business process model. It explicitly states a result of the activity in its domain of execution. Effect annotations are formed in the indicative mood, or as facts (e.g. 'A courier has provided an unsigned contract to a partner organization.'). A *causal* relationship exists between a process activity and an effect. An activity can cause many effects, and an effect can be caused by many activities. Effects can be viewed as both: *normative* - as they state the required outcomes; and, *descriptive* - in that they describe the normal, and predicted, subset of all possible outcomes (i.e. actual outcomes may vary at run-time).

Effect annotations can be *formal* (for instance, in first order logic, possibly augmented with temporal operators), or *informal* (such as simple English). We recommend in both cases, that *informal* annotations of effect be applied as a first pass to ensure a rich expression, and for ease of communication. Many of the examples we use in this paper rely on informal natural language effect annotations. Ideally, and for analysis purposes, it would be of benefit if the annotations were stated formally as this permit us to use automated reasoners, while informal annotations oblige analysts to check for consistency and completeness between effects (as discussed below). A middle-ground can be worked when effect annotations are formed using some predefined form that can be translated automatically into a formal representation. For example, via the use of Controlled Natural Languages (CNL) with grammar and vocabulary restrictions such as in Schwitter (1996), and Sowa (2004).

When an analyst is annotating existing process models, the conditions labelling control-flows leaving a decision gateway may provide some understanding of the effect of a downstream activity. Effects may also refer to assumptions on how the immediate state of an observer (i.e. during process inter-operation across pools) may change as a result of some information / work item transfer. When implemented within a tool, effects may be viewed on a business process model graphically, or added to meta-information of activities or sub-processes.

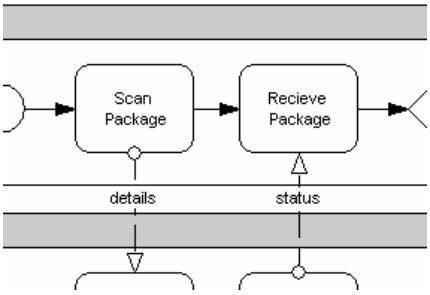
An annotated BPMN model, for the purposes of this paper, is one in which every task (atomic, loop, compensatory or multi-instance) and every sub-process has been annotated with descriptions of its immediate effects. We will now describe a methodology for accumulating these effect annotations to obtain a cumulative

effect annotation for a complete process. We will assume that informal annotations are available in describing this methodology.

**Effect Accumulation**

Effect annotations are statements concerning the immediate effect of a particular task. In order to identify the cumulative effect of a complete process, we combine the effect of tasks executed in a pair-wise manner. This provides the analyst with a cumulative effect as the accumulation is progressed through traversal of the activities in the process. This accumulation equates to stating that all (or some, as will be discussed below) of the prior effects 'AND' the immediate effects of the task to receive the cumulative effect, are true at the cumulative point in the process. That is, when given an ordered pair of tasks with effect annotations, the accumulation determines the cumulative effect after both tasks have been executed in contiguous sequence. Pair-wise effect accumulation only occurs across control-flow links between tasks within participant lanes.

Take Figure 4 as an example. Let a task $T_1$ be the preceding task in the sequence (i.e. 'Scan Package'), and $T_2$ be the succeeding task (i.e. 'Receive Package'). The cumulative effect of the process at $T_2$ results from combining its immediate effects with the cumulative effect of its preceding task.




*EffectScenaroT*$_1$ = 'All packages have been scanned.' AND 'Some packages have not been screened.' …

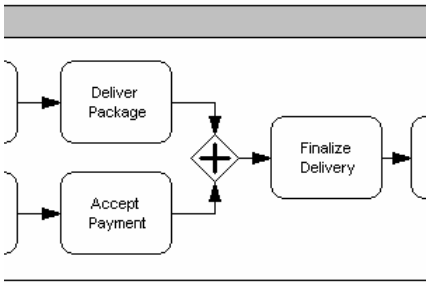
*EffectScenarioT*$_2$ = 'All packages have been screened.' AND 'Some packages have been received.' …

*CumulativeEffectT*$_2$ = 'All packages have been scanned.' AND 'All packages have been screened.' AND 'Some packages have been received.' …

Figure 4: Pair-wise accumulation of two tasks in contiguous sequence.

During process enactment, the effects of a task may override the effects of a preceding task. For example, say an effect in $T_1$ states 'Some packages have not been screened', and an effect in $T_2$ states 'All packages have been screened'. In this case, the effect of $T_2$ will override the effect of $T_1$. The effects that require an override can be identified by searching for any contradictions in the effects to be accumulated in the prior task in the sequence, given the immediate effects of the current task. This will result in the inclusion of as many of the effects in the cumulative effect of the prior task $T_1$ that are not contradictory, when accumulated to the succeeding task $T_2$. The process continues without modification over splits. Joins require special consideration. These result in alternate *effect scenarios* when XOR-joins or OR-joins have been used (as will be described below). In the following, we describe the methods to be followed in the case of 2-way joins only, for brevity. These methods generalize in a straightforward manner for *n*-way joins.

Figure 5 represents part of a process that includes an AND-join where $T_1$ = 'Deliver Package', $T_2$ = 'Accept Payment', and $T_3$ = 'Finalize Delivery'. Firstly we accumulate the immediate effects of $T_3$ with the cumulative effects of both $T_1$ and $T_2$. The immediate effects of $T_3$ are combined with all alternate effect scenarios that have been accumulated on either $T_1$ or $T_2$. This yields the sets of cumulative effects $T_{1+3}$ and $T_{2+3}$. This accumulation includes the analysis of any contradictions as previously discussed, which takes any overrides into consideration. We then combine $T_{1+3}$ and $T_{2+3}$ to signify the cumulative effect at $T_3$. In this case, any effect scenarios accumulated on either $T_1$ or $T_2$ remain, with additional effects as per the immediate effects of $T_3$. Note that we do not consider the possibility of a pair of effect scenarios having any contradictions, since this would only happen in the case of intrinsically and obviously erroneously constructed process models.




*EffectScenaroT*$_1$ = 'Some packages have been delivered to a customer.' …

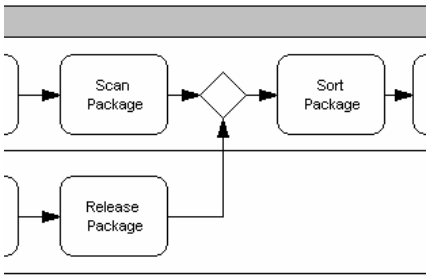
*EffectScenarioT*$_2$ = 'Full payment has been received from a customer for some packages that have been delivered.' …

*EffectScenarioT*$_3$ = 'The delivery of some packages is finalized.' …

*CumulativeEffectT*$_3$ = 'Some packages have been delivered to a customer.' AND 'Full payment has been received from a customer for some packages that have been delivered' AND 'The delivery of some packages is finalized'…

Figure 5: Pair-wise accumulation to a task during an AND-join.

In Figure 6, an XOR-join is represented, where either task $T_1$ = 'Scan Package', or $T_2$ = 'Release Package' have executed and task $T_3$ = 'Sort Package' has completed during enactment. In this case, two effect scenarios are to be generated at $T_3$ with the cumulative effects of $T_1$ and $T_2$ respectively. Firstly we accumulate the immediate effects of $T_3$ with the cumulative effects of both $T_1$ and $T_2$. The immediate effects of $T_3$ are combined with all alternate effect scenarios that have been accumulated on either $T_1$ or $T_2$ to yield the cumulative effects $T_{1+3}$ and $T_{2+3}$. Any overrides are applied (as previously discussed). $T_{1+3}$ and $T_{2+3}$ then remain in the cumulative effect at $T_3$ as alternate effect scenarios.




*EffectScenaroT*$_1$ = 'Some packages have been scanned at some sort facility.' …

*EffectScenarioT*$_2$ = 'Some packages are released from some bonding area.' …

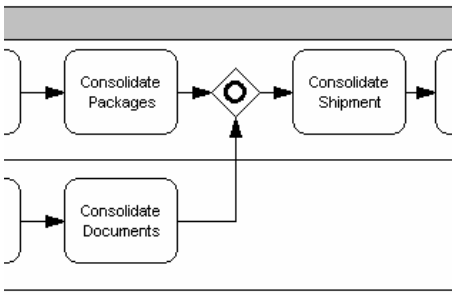
*EffectScenarioT*$_3$ = 'All packages have been sorted to their destination routing facility' …

*CumuEffScen*$_1$*T*$_3$ = 'Some packages have been scanned at some sort facility.' AND 'All packages have been sorted to their destination routing facility'…

*CumuEffScen*$_2$*T*$_3$ = 'Some packages are released from some bonding area.' AND 'All packages have been sorted to their destination routing facility'…

Figure 6: Pair-wise accumulation to a task during an XOR-join.

Finally, Figure 7 represents an OR-join, with tasks $T_1$ = 'Consolidate Packages', $T_2$ = 'Consolidate Documents', and $T_3$ = 'Consolidate Containers'. In this case, either: $T_1$ 'AND' $T_2$ may have completed; or, $T_1$ 'OR' $T_2$, prior to the completion of $T_3$. Therefore, we combine the methods for AND-joins, and XOR-joins to identify the cumulative effect at $T_3$. Firstly we determine the result in the scenario where both $T_1$ 'AND' $T_2$ have completed (i.e. as in the AND-join previously discussed). We then determine the result whereby either $T_1$ 'OR' $T_2$ have completed (i.e. providing alternate effect scenarios). We then combine both results to provide the cumulative effect at $T_3$. That is, if there were only one effect scenario for both $T_1$ and $T_2$, the result will be three cumulative effect scenarios at $T_3$.




*EffectScenaroT*$_1$ = 'Some packages have been consolidated into containers.' …

*EffectScenarioT*$_2$ = 'Some documents have been consolidated into bags.' …

*EffectScenarioT*$_3$ = 'All shipments have been consolidated for delivery.' …

*CumuEffScen*$_1$*T*$_3$ = 'Some packages have been consolidated into containers.' AND 'All shipments have been consolidated for delivery.' …

*CumuEffScen*$_2$*T*$_3$ = 'Some documents have been consolidated into bags.' AND 'All shipments have been consolidated for delivery.' …

*CumuEffScen*$_3$*T*$_3$ = 'Some packages have been consolidated into containers.' AND 'Some documents have been consolidated into bags.' AND 'All shipments have been consolidated for delivery.' …

Figure 7: Pair-wise accumulation to a task during an OR-join.

**Fulfilment Conditions**

A *fulfilment condition* (Fuxman et. al 2004) is a statement specifying the required conditions realized upon completion of a given task, goal or dependency in an organizational model (an *i** model in this case). Fulfilment conditions recognize the required effects on a business process model. For example, a fulfilment condition for a task dependency to 'BondAPackage', may include an effect stating: 'Some packages have been forwarded to some bond facility '. Fulfilment conditions annotated to dependencies will intuitively be required by the task the dependency is linked to on the dependee actor. This implies that the dependee task must include the capability to realize the fulfilment conditions of any of the dependencies it is required to fulfil.

Fulfilment conditions are annotated in the same manner as effects in business process models. In *i**, fulfilment conditions are annotated to tasks and goals assigned to actors in an SR diagram and dependencies in an SD model. At this point in time, we do not include soft-goals during annotation as they describe non-functional properties used during assessment of alternative structures.

**Establishing Realization Relationships between Elements of both Organization and Business Process Models**

To effectively manage change, we need to deal with changes to both the processes themselves and also to the organizational context. In both cases, we need to evaluate the impact of these changes on the process model with

reference to models of the organizational context. Impact can be determined via an assessment of *realization* between elements represented within organizational models and business processes.

We establish true realization by first establishing *normative realization* links between a BPMN model and an *i** model. Such links relate activities or sub-processes, to tasks in an *i** model. A normative realization link must be established by an analyst and suggests that the task in question must ideally be realized by the process it is linked to. We determine whether this normative statement actually describes reality over two steps (described below).

We establish *weak realization* of the link, which determines whether the effects of the task-process pair are contradictory. Contradictory effects preclude the possibility of the process realizing the task, while consistent effects and therefore weak realization leaves open such a possibility. Identification of weak realization is similar to the process discussed during effect accumulation, whereby any contradictory effects are identified. In this case however, any contradictions between fulfilment conditions and effect scenarios signify that the required fulfilment conditions defined in the organizational model *cannot be realized* during process execution. This is due to an *inconsistency* between the effects of the process, and the fulfilment conditions. For example, a fulfilment condition requiring 'All participants have agreed to the proposed date' is inconsistent with an effect stating 'Some participants have not agreed to the proposed date'. In the following, we will refer to object-level consistency between assertions of effects and fulfilment conditions. We will also refer to consistency between models and consistency labels on normative realization links. The context of use will clarify which specific notion of consistency is being used in each instance. When only informal annotations are available, consistency checking involves analysts evaluating natural language descriptions of effects and fulfilment conditions to determine if they are contradictory.

We then establish *true realization* to indicate that the process in question does indeed realize the task that it is linked to. True realization is established by identifying whether each process related fulfilment condition is *entailed* in each effect scenario in the cumulative effect of the process. This involves analysts evaluating whether the fulfilment conditions annotated to the task in the organizational model always hold when the effects are true. That is the fulfilment conditions *follow on from* the effects. In this case, the analyst is to be provided with the cumulative effect scenarios resulting from effect accumulation of the process. The analyst then assesses each fulfilment conditions against each effect scenario in the cumulative effect of the process. True realization is established if the analyst can identify that the fulfilment conditions hold in each effect scenario.

**Assessing Realization between Organizational and Business Process Models**

We now describe a methodology for assessing the level of realization between an organizational model (as represented in *i**) and a business process model (as presented in BPMN).

The methodology is to be followed in determining realization between a BPMN model and an *i** model. Note that we will label links as weakly/truly realized/unrealized – this is merely for convenience, bearing in mind that the corresponding labels actually describe consistency/inconsistency of the elements that are related via these links. Note also that consistency and entailment checking between effects and fulfilment conditions can be automated with the use of formal reasoners when formal annotations are available (we require that all effect annotations and fulfilment conditions are specified in the same formal language).

*Step 1:* A set of *normative realization links* between the BPMN model and the *i** model is established by an analyst. Tasks in an *i** model are to be normatively linked to activities and sub-processes in a BPMN model. The internal structure of tasks in an *i** model provides some guidance for establishing links. Lower level tasks should conceivably be represented as either sub-processes or activities in a BPMN model that is linked to some higher level task. This may not be the case however, where a sub-task in an *i** model has been represented at some lower level of decomposed detail in a BPMN model.

*Step 2:* For each such link:

a. We first determine *weak realization*. A normative realization link is deemed to be weakly realized if every effect scenario in the cumulative effect of the process is consistent with the fulfilment conditions of the corresponding task in the *i** model. That is, the analyst has reviewed each effect scenario against the fulfilment conditions in the *i** model, and can safely say that there are no contradictions between effects. The link is labelled *unrealized* otherwise.

b. We next determine *true realization*. A normative realization link is deemed to be truly realized if:

i. It is *weakly realized*, and

ii. Each fulfilment condition of the task in question is entailed (as discussed in the previous section) by each effect scenario, in the cumulative effect of the process in question.

iii. Otherwise, the link is deemed to be *unrealized*.

*Step 3:* Given a process model, an organizational model and a set of normative realization links relating the two:

a. The process model is said to be *unrealized* (or inconsistent) with the organizational model if there exists *at least one* normative realization link that is deemed to be unrealized.

b. The process model is said to be *weakly realized* with the organizational model if all normative realization links are deemed to be weakly realized. Otherwise, the models are weakly unrealized.

c. The process model is said to be *truly realized* with the organizational model if all normative realization links are deemed to be truly realized. Otherwise, the models are truly unrealized.

Note that true realization implies weak realization, both for normative realization links and for pairs of models.

## MANAGING PROCESS PORTFOLIOS WITH ARCHITECTURAL MODELS OF AN ORGANIZATION

In previous sections we have shown how relationships can be established between business process and organisational models. We have also discussed the nature of change and identified that change may either occur at an organizational or business process level. We now discuss how the change can be effectively supported by the methodology we have established.

We propose the use of organizational models to provide the 'scaffolding' with which to organize business processes. This allows for improved traceability to the greater organizational context via explicit representation for issues of strategic importance such as participant motivation and inter-dependency. Thus the framework, illustrated in Figure 8, can be effectively used to manage the "entire set of processes" in a more holistic manner.

A *Process Portfolio* is "…a coherent treatment of the entire set of processes, allowing them to be improved in total, rather than streamlining one and, consequently, unknowingly, suboptimizing others" (Rosemann, 2006). It provides a holistic view of an organizations processes, their relationships and salient properties. Some of the proposed contributions of process portfolio management include: an ability to "…provide an initial structure in a process-unaware organization"; improved support for understanding "…the most important set of its business processes" according to their qualitative aspects such as risk, criticality, impact and opportunity etc; and the utilization of "…not only process model data, but also corresponding information about the actual process executions" during improvement efforts.

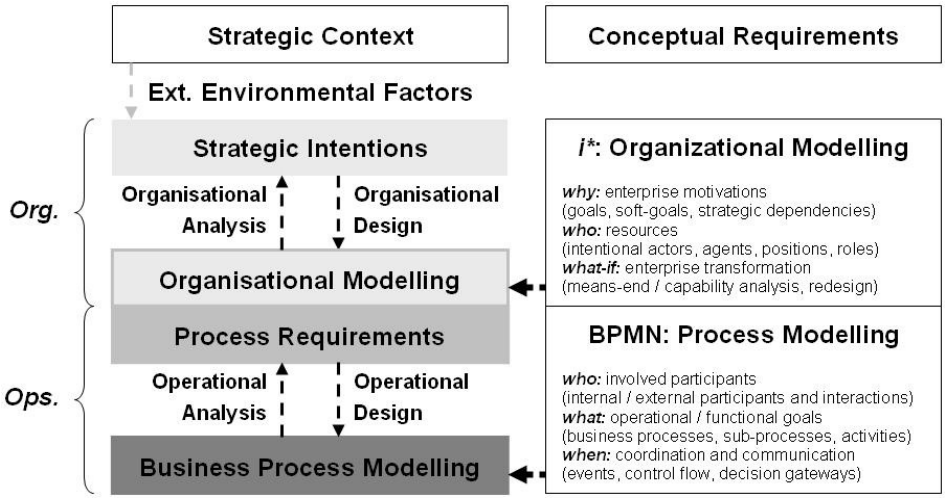



Figure 8: Enterprise modelling framework.

The *i** framework's focus is on the strategic relationships between organizational actors and their underlying motivations and capabilities. As such, it provides an ideal high-level representation framework for initial process elicitation. Specifically, its sequence agnostic characteristics allow an analyst to focus on architectural requirements from an intentional actor perspective, and leave operational requirements such as co-ordination and communication for the later phases of detailed design. Coupled with our annotations and methodology, any processes constructed can then be verified against organizational requirements. In this sense, once a valid architecture is agreed upon, process design and construction becomes a significantly easier task. In addition, any subsequent changes post-design can then be assessed against their impact at the organizational level, as will be illustrated (with examples) below.

**Assessing the Impact of Process Change with reference to an Organizational Model**

The process in Figure 10 represents the 'Bond Package' sub-process in Figure 3, that has been changed as per "Change 1: Task/Flow Addition". Previously true realization links were established between the 'Bond Package' sub-process and the tasks to 'Receive[Package]', 'Bond[Package]' and 'Provide[Details]' for the organizational model illustrated in Figure 2. Their fulfilment conditions are represented in Table 9.

| Task | Fulfilment Conditions |
|---|---|
| *Receive[Package]* | 'Some Packages have been received at the bond facility.' AND 'All received packages have been scanned.' |
| *Bond[Package]* | 'All received packages have been bonded at a bond facility.' |
| *Provide[Details]* | 'The details for all received packages have been provided to the regulatory authority.' AND 'All received packages have been screened'. |

Table 9: Associated tasks and effects for the Bond Package process in Figure 10.

We initially describe and assess the first change that resulted in the model in Figure 10, and then describe and assess another proposed change – "Change 2: Task Removal".

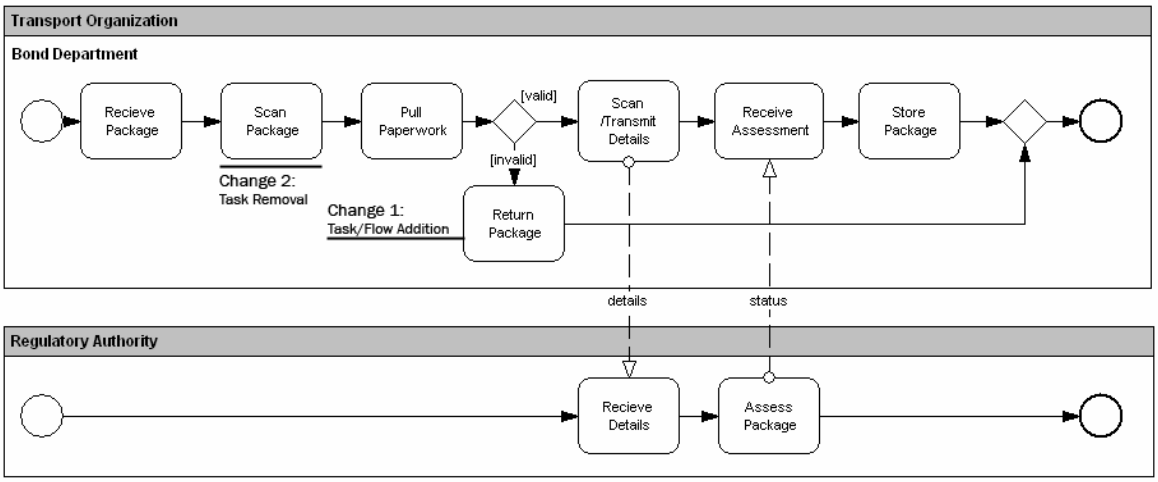



Figure 10: Assessing the impact of process change in the Package Routing example.

*Change 1:* In this case, the change involves the introduction of a new task 'Return Package' and exclusive decision gateway 'Valid/Invalid' (see Figure 10). The intent of this change was to reduce the processing time within the bond, by re-routing packages with invalid paperwork (i.e. unreadable, or for another destination / organization) back to the sort facility that initially forwarded the package. The 'Return Package' task is also annotated with: 'The details for some received packages have not been provided to the regulatory authority' effect. The result of effect accumulation is two effect scenarios, with one containing the aforementioned effect. Upon analysis, the new effect scenario introduces a contradiction with the fulfilment conditions of the 'Provide[Details]' task, denying its realization. By analysing the organizational model (Figure 2), the fulfilment condition can be traced back to the 'Package Details' resource dependency the Regulatory Authority requires to 'Handle[Package Clearance]'. In summary, this change has reduced the realization relationship between the models to *unrealized,* due to the contradiction.

*Change 2:* This change concerns the proposed removal of the 'Scan Package' task in Figure 10. The task includes an effect stating: 'All received packages have been scanned', that realizes the a fulfilment condition annotated to the 'Receive[Package]' task in the organizational model. This change has the intent to reduce the package handling requirements of the Bond Department on the assumption that the Sort Facility will be scanning the packages prior their receipt. Upon accumulation the effect is not stated as being realized by the process, therefore the status of realization is reduced to weakly realized. That is, it is not explicitly stated whether the effect has or has not been realized.

Both the above changes illustrate simple scenarios of where the effect annotations and organizational model may be used to effectively manage process portfolios and change. In the first case, a change may have resulted in a process that is uncompliant with regulatory constraints. In the second case, a weaker reduction in the realization of process requirements was the case. In both cases, significant operational impact may still be foreseeable. However, the traceability available in the methodology and annotations provides a first step towards resolving

actual / possible inconsistencies at design time, even in environments where processes are independently designed and managed.

## CONCLUSION

In this work, we have provided a method to support effective process change against higher-level models of the organizational context. Process change occurring during design can be connected to elements on the organizational model to be realized. Once connections are established, the fulfilment conditions of related organizational elements serve as requirements to be considered during process design. In addition, any changes at an organizational level can then be traced to elements at the process level for analysis of current capabilities, possibly triggering / focussing improvement efforts. Ideally, the methods and extensions are to support process design, and as such require integration with an associated tool. We are actively pursuing this task and hope to provide applied field results in the near future.

## REFERENCES

Aalst, W.M.P van der, (2003) Inheritance of business processes: A journey visiting four notorious problems. Petri Net Technology for Communication Based Systems, *Lecture Notes in Computer Science,* Springer-Verlag, Berlin 2678 (2003) 383–408.

Becker, J. Indulska, M. and Rosemann, M. Green, P. (2005) Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN, in Campbell, Bruce and Underwood, Jim and Bunker, Deborah, Eds. *Proceedings 16th Australasian Conference on Information Systems,* Sydney, Australia, 2005.

Fischer, L. (2005) *Workflow Handbook 2005,* Workflow Management Coalition, (WfMC), 2005.

Fuxman, A. Liu, L. Mylopoulos, J. Pistore, M. Roveri, M. Traverso, P. (2004) Specifying and analyzing early requirements in Tropos. In: *Requirements Engineering*, Springer London, 9(2) (2004) 132–150.

Hammer, M. Champy, J. (1993) *Reengineering the Corporation: A Manifesto for Business Revolution,* HarperBusiness, 1993.

Harmon, P. (2003) *Business Process Change,* San Francisco, CA: Morgan Kaufmann, 2003.

Ouyang, C. W.M.P. van der Aalst, Dumas, M. and ter Hofstede, A.H.M. (2006) *Translating BPMN to BPEL*, BPM Center Report BPM-06-02, BPMcenter.org, 2006.

Schwitter, R., Fuchs, N. (1996) *Attempto - from specifications in controlled natural language towards executable specifications.* EMISA Workshop 'Naturlichspraclicher Entwurf von Informationssystemen' (1996)

Smith, H. Fingar, P. (2003) *Business Process Management – The Third Wave,* Tampa, FL: Meghan-Kiffer Press, 2003.

Sowa, J.F. (2004) *Common logic controlled english (draft).* Technical report, http://www.jfsowa.com/clce /specs.htm Feb, 2004.

Rosemann, M. (2006) Process Portfolio Management, URL http://www.businessprocesstrends.com, Accessed 6 Aug 2006

White, S. (2006), Business Process Modeling Notation (BPMN), Version 1.0, OMG Final Adopted Specification, February 6, 2006.

Yu, E. (1995) *Modelling Strategic Relationships for Process Reengineering,* PhD Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, Canada, pp.~124, 1995.

## COPYRIGHT

George Koliadis, Aditya Ghose & Moshiur Bhuiyan © 2006. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.