1-1-2007

# Relating business process models to goal-oriented requirements models in KAOS

George Koliadis
*University of Wollongong*, gk56@uowmail.edu.au

Aditya K. Ghose
*University of Wollongong*, aditya@uow.edu.au

### Abstract
Business Process Management (BPM) has many anticipated benefits including accelerated process improvement, at the operational level, with the use of highly configurable and adaptive "process aware" information systems [1] [2]. The facility for improved agility fosters the need for continual measurement and control of business processes to assess and manage their effective evolution, in-line with organizational objectives. This paper proposes the GoalBPM methodology for relating business process models (modeled using BPMN) to high-level stakeholder goals (modeled using KAOS). We propose informal (manual) techniques (with likely future formalism) for establishing and verifying this relationship, even in dynamic environments where essential alterations to organizational goals and/or process constantly emerge.

# Relating Business Process Models to Goal-Oriented Requirements Models in KAOS (Preprint)

George Koliadis[1] and Aditya Ghose[1]

School of Information Technology and Computer Science
University of Wollongong
Wollongong, N.S.W. Australia
{gk56, aditya}@uow.edu.au
http://www.uow.edu.au

**Abstract.** Business Process Management (BPM) has many anticipated benefits including accelerated process improvement, at the operational level, with the use of highly configurable and adaptive "process aware" information systems [1] [2]. The facility for improved *agility* fosters the need for continual *measurement* and *control* of business processes to assess and manage their effective evolution, in-line with organizational objectives. This paper proposes the GoalBPM methodology for relating business process models (modeled using BPMN) to high-level stakeholder goals (modeled using KAOS). We propose informal (manual) techniques (with likely future formalism) for establishing and verifying this relationship, even in dynamic environments where essential alterations to organizational goals and/or process constantly emerge.

## 1 Introduction

Business Process Management (BPM) in its "third wave" [1] has been conveyed as: enabling intelligent business management [3]; facilitating the redesign and organic growth of information systems [4]; and, obliterating the business - IT divide [1]. Business processes undergo an evolutionary life-cycle of change. This change is brought on by the need to satisfy the constantly changing goals of varied stakeholders and adapt to the accelerating nature of change in today's business environment [5]. The need for change is best described in [6] as the transition from an initial "unsatisfactory" (i.e. as-is) state to a new hypothetically "desired" (i.e. to-be) state. The desired state is theoretically based on the assumption that it more effectively satisfies related operational goals [4] [7] [8] in-line with higher-level strategic goals. It is therefore important that the criterion for effective process change - i.e. stakeholder goals, be explicitly stated, communicated and traceable to any changes that are proposed, approved, and/or implemented.

The new-found *agility* provided by BPM, however presents the need for methods to successfully *control* and *trace* the evolution of processes. This need is affirmed in [7], by stating that organizations evolve from their original intentions through complex and unpredictable growth. BPM aims to support the evolution of organizations and their processes, however controls are still needed to ensure that operational as well as higher-level goals (i.e. of more strategic concern) are continually satisfied, allowing for "organizational growth in the right direction". In order to meet goals however, there is a need to support traceability between processes and organizational goals - "You can't manage what you can't trace" [9].

We have proposed a method (GoalBPM) to support the controlled evolution of business processes. Control is supported through the explicit modeling of stakeholder goals, their relationships (be it either refinement, conflict or obstruction), and their evolution traceable to related business processes. GoalBPM is used to couple an existing and well-developed, informal-formal goal modeling and reasoning methodology - i.e. KAOS [10], and a newly developed business process modeling notation - i.e. BPMN [11]. This is achieved through the identification of a *satisfaction* relationship between the concepts represented. GoalBPM itself can be seen as an "adapter" that integrates the two models, to support their co-evolution and synergistic use.

This paper firstly presents a background to the associated domains of business process modeling and goal-oriented requirements engineering. An informal overview of the GoalBPM method is subsequently outlined with a simple example for illustration.

## 2 Business Process Modeling with BPMN

We have initially chosen the Business Process Modeling Notation (BPMN), developed by the Business Process Management Initiative (BPMI.org) for use in the construction of GoalBPM.

A *Business Process* is a set of dynamically co-ordinated activities controlled by a number of dependent, social participants. Processes are represented in BPMN using **flow objects**: *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 boxes containing a single process), and *lanes* within pools (sub-boxes).

We refer to Figure 2, a public Package Sorting process, as an example to illustrate BPMN. The process requires the interaction of two high-level process participants - the Transport Organization, and a Transport Authority. Collaboration between participants on the model is represented by *message flow links* between *activities* within *pools*. Responsibility within the Transport Organisation is delegated to two roles - the Sort Operations, and a Bond Operations. Responsibility assignment within a pool is represented using *lanes* (i.e. pool divisions). Each pool within a process model represents a single process. Processes are initiated by a *start event*, represented as a circle at the beginning of each pool. *Control flow links* between activities, decisions and events, represent the controlled progression through each process. A *decision gateway*, (i.e. diamond) can be seen in the figure, identifying the need to make a choice on whether to bond a package. Finally, the process is completed with an *end event*, or bold circle toward the end of a process.

## 3 Goal-Oriented Requirements Engineering with KAOS

We have chosen a GORE method that is focused toward both the early and late phases of RE, specifically KAOS (Knowledge Acquisition in autOmated Specification of software systems) [10], to represent the organizational goals related to business process execution.

**Goal Declaration in KAOS** Goals are declared in terms of desired, timely effects within a composite system (e.g. *Achieve*[*MeetingScheduled*]). Goals are conceptually modeled in KAOS on a semantic net that represents a hierarchy of parent goals and their refinements into sub-goals. Goals that exist higher in the hierarchy represent the high-level goals (i.e. *strategic* concerns) of the organization. These goals are not "clear cut", in the sense that their satisfaction is complex and cannot be proven without common interpretation. These goals are then *refined* down the hierarchy into sub-goals that are more *operational* in nature. That is, their assignment to a small group of individuals responsible for a number of operational activities illustrates the means by which they are satisfied.

Goals can be either 'AND' or 'OR' refined (Figure 3). An 'AND' refinement of a goal states that the parent goal is satisfied if all the goals in the refinement are satisfied. An 'OR' refinement on the other hand states that the parent goal is satisfied if a single refinement is satisfied. This allows for the modeling of alternative refinements for goal satisfaction. KAOS also provides a criterion for halting the refinement process, in that if a goal can be assigned the sole responsibility of a single environmental role (i.e. agent in the composite system); there is no need for further goal refinement to occur. This also provides a means by which to make the transition between goals and the constrained operations that satisfy those goals [12].

**Goal Definition in KAOS** KAOS supplies an optional formal assertion layer that allows for the specification of goals in Real-Time Linear Temporal Logic (RT-LTL) [10] [13]. These formal goal assertions allow for precise specification of goals, as well as supporting the use of developed formal reasoning techniques that aid in identifying/resolving conflicts between goals and proving absolute/partial goal satisfaction.

A formal goal definition in KAOS begins with the assertion of the objects the goal *concerns*. In KAOS, these objects are declared in the object model. The definition then states the desired temporal ordering of states the concerned objects must hold in order to *satisfy* the goal.

Goals are defined in the form:

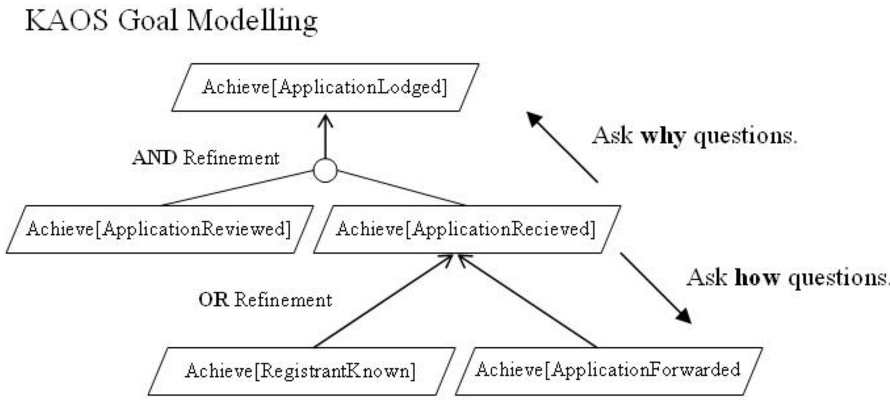


**Fig. 1.** Modeling goals in KAOS

C ⇒ *op*T,

where C and T are assertions about environmental situations (i.e. current and target), and *op* is a temporal operator that signifies the desired temporal nature of the target situation (T), in relation to a current situation (C). This paper requires knowledge only of the two most used operators. Namely, ‘at *some* time in the future’ denoted by a open diamond ‘⋄’, and ‘at *all* times in the future’ denoted by an open square ‘’. For a complete list of the operators used in KAOS, see [10].

Take for example, the ‘PackageSortedToDestination’ goal below:

**Goal** Achieve[PackageSortedToDestination]
**InformalDef** If a package is received at a sort facility, then the package will eventually be forwarded to its known destination.
**FormalDef** ∀p: Package, sf: SortFacility
Received(p, sf) ⇒ ⋄Forwarded(p, p.Destination)

Also note: package destinations (i.e. p.Destination) are *attributes* of a package.

**Patterns for Declaring and Defining Goals** KAOS defines a number of commonly used “Goal Patterns” that generalize the timeliness of target situations. They provide an informal method to initially declare goals, as well as to guide formal definition. *Achieve Goals* ($C \Rightarrow \diamond T$) desire achievement ‘some time in the future’. That is, the target must eventually occur. (e.g. Achieve[PaymentRecieved]). *Cease Goals* ($C \Rightarrow \diamond - T$) disallow achievement ‘some time in the future’. That is, there must be a state in the future where the target does not occur (e.g. Cease[Operation]). *Maintain Goals* ($C \Rightarrow T$) must hold ‘at all times in the future’ (e.g. Maintain[EmployeeSafety]). *Avoid Goals* ($C \Rightarrow -T$) must not hold ‘at all times in the future’ (e.g. Avoid[LateEntry]).

## 4 Linking Goal and Process Models

The GoalBPM methodology relies on establishing relationships between goal models and process models. This relationship is established in two stages. First, *traceability links* are established between goal nodes in the goal model to activities (or end events of complete processes) in the process model. Second, *satisfaction links* are established between goals and processes.

A traceability link is an informal statement of a relationship between a goal and a process (or sub-process). In establishing such a link, we are effectively asserting that the goal in question has some bearing on the process (or sub-process) under consideration. A traceability link does not necessarily lead to a satisfaction link. Sometimes, a process or sub-process may be related to a goal because it *obstructs* it, in the sense of [14]. We draw a traceability link between a goal and a process end event if the entire process in question has some bearing on the goal. We draw a traceability link between a goal and an activity if the sub-process ending in that activity relates to the goal. In general, traceability links need to be established by analysts. Some guidance can be offered in this process by using cues present in the goal and process models. For instance, the names of goals and processes (or activities) can suggest traceability links. A transport organization's operational goal to achieve "PackageSortedToDestination" can be traceable to the "Package Sorting" process within the organization.

Traceability between goals and processes can be identified through cross examination of the links between the pre/post conditions for specific processes and the pre/post conditions for specific goals. A process is made available for execution when a specific pre condition has been met (e.g. a *customer that has submit a registration form* in the 'Register New Customer' process), and completed upon meeting a post condition (e.g. the customer is validated and their details are stored). These pre and post conditions can be related to the pre and post conditions for specific organizational goals. Take for example the goal, '*all new customer registrations* require credit reporting and verification', is related to the 'Register New Customer' process by way of the pre condition.

A *satisfaction link* is a traceability link where the process (or sub-process) in question satisfies the goal involved in the link. A satisfaction link can be of two types:

- *Normative satisfaction links:* These indicate that a process or sub-process *must* satisfy the relevant goal. Such links articulate desired states of affairs.
- *Descriptive satisfaction links:* These indicate the "as-is". We obtain descriptive satisfaction links from effect annotations of processes, using techniques that we discuss in the next section.

## 5 Using Model Annotations to Verify Goal Satisfaction

GoalBPM establishes satisfaction links between goals and process models in three steps. First, it annotates process models with *effect annotations*. Second, it identifies a set of *critical trajectories* from a process model. Third, it identifies the subset of the set of traceability links that represent satisfaction links by analyzing critical trajectories relative to process effect annotations. The satisfaction links thus obtained are *descriptive* satisfaction links. A final step in GoalBPM is to use a comparison of the set of normative satisfaction links with the set of descriptive satisfaction links to drive the processes of *goal model update* and/or *process model update*.

Our approach may be viewed as an instance of the *state-oriented* view [15] [16] [17] of business processes as opposed to the *agent-oriented* or *workflow* views. However, we are not explicitly state-based in that we do not seek to obtain state machine models from process models, for two reasons. First, BPMN models in general do not guarantee finite state systems, making the application model checking techniques difficult. Second, the derivation of state models from BPMN models appears difficult at this time, due to the high-level, abstract nature of BPMN models.

### 5.1 Effect Annotations

A *process activity* (i.e. as represented in BPMN as a rounded box) is an element on a process model that indicates required state transitions in order for the process to progress toward the achievement of all related goals. The labeling of an activity (e.g. 'Register New Customer') generalizes a number of possibly desired/undesired results, or outcomes. Due to this generalization, most process models do not satisfactorily depict the lowest level state achievements required for process progression. They are too 'high-level' and do not provide an in depth understanding of the process and its ability to achieve desired goals or objectives. This understanding is important when trying to prove goal achievement, which is reliant on the achievement of certain target states. In order to provide greater understanding of process models (i.e. state transitions in particular) to support their analysis in relation to the goals they hope to satisfy, we augment the process model with 'effect annotations'.

An *effect* is the result (i.e. product or outcome) of an activity being executed by some cause or agent. It indicates the achievement of a certain environmental state communicated through an event. 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 if the conceptual model were to be hypothetically executed. A *cause* relationship exists between a process activity and an effect (i.e. the process activity causes the effect to occur). An activity can cause many effects and an effect can be caused by a number of activities.

The manner in which an activity is executed may result in alternative outcomes. For this reason, effect annotations are related to activities in an AND/OR refinement similar to goal refinement in KAOS. This alternative execution is derived from the dynamic co-ordination represented in the process model through decision gateways (i.e. diamonds in BPMN). Decisions commit to alternate paths of execution based on the current state of the process. This is achieved through the specific effects of prior activities that have been executed. The conditions for the decision on which choice of path to commit to can help to identify important effects on prior activities in the current, or in other processes. These influential activities and their required effects for the current path of execution, need to be identified and represented along with the effects of the current process to prove goal satisfaction.

We define an *effect annotation* to include:

– a *label* that generalizes the behavior of the effect in relation to its environment (e.g. 'CustomerDetailsStored'). Whereas the labeling of an activity is made in the optative mood (i.e. a desire), an effect annotation is made in the indicative mood (i.e. a fact).ling of an activity is made in the optative mood (i.e. a desire), an effect annotation is made in the indicative mood (i.e. a fact).
– a *designation* specifying whether the effect is a 'normal' (i.e. expected) outcome for the activity that in turn aims toward goal achievement, or an 'exceptional' (i.e. unaccepted) effect that deviates from goal achievement. (e.g. 'RegistrationValidated' may be a normal outcome for a customer registration activity, whereas 'RegistrationRejected' may be exceptional)
– a *informal definition* an informal definition describing the effect in relation to the result achieved in its environment (e.g. 'The details relating to the current customer have been stored within the system.'). This provides an informal explanation (i.e. meaning) of the effect in relation to the real-world environment.
– a *formal definition* (optional) defining achieved states to aid in mapping to formal goal definitions in the chosen goal definition formalism (i.e. in this case KAOS). (e.g. '∀ c: Customer, (∃ cr: CustomerRecord) Stored(c.Details, cr)')

At the tool level, effect annotations can be viewed on a business process model graphically, or added to meta-information relating to the process activities. They can then be analyzed along with the process and associated goals as described in the subsequent sections.

### 5.2 Trajectory Decomposition

The dynamic co-ordination of activities controlled by responsible agents is represented / supported on a business process model by way of decision gateways (i.e. diamonds in BPMN). This manner of 'per instance' process control allows for the existence of many *process trajectories* through the process (i.e. possibly even an infinite number when cycles are included). We classify a single trajectory as a unique and supported sequence of activity execution. Each trajectory results in a 'cumulative effect' for the given process. We use the term trajectory to signify a specific 'chosen path' through the process that results in a specific/unique outcome or 'cumulative effect'.

During the trajectory decomposition process, specific effects need to be chosen where alternative effects are available on each activity. This choice relies on the decisions influencing the path of the particular trajectory. We choose between alternative effects based on their conformance to the current trajectory.

Business process models support and represent exceptional trajectories. These trajectories do not necessarily satisfy process goals. They react to exceptional events that occur in the process by re-routing the current path of execution so that alternative steps can be taken to either resolve the exceptional situation or abort the process. We can therefore label a trajectory as having either a *normal* or *exceptional* type, based on its final achieved state. Commonly, a trajectory that cannot resolve exceptions and requires termination prior to meeting all the goals it must satisfy, is classed as exceptional. This is discussed in the following section.

The identification of all unique process trajectories can be difficult, given the complex nature of activity interleaving and iteration possible in the process models. For this purpose, it is recommended that an automated method for deriving all possible process trajectories be available.

### 5.3 Goal Satisfaction and Cumulative Effect Assessment

We firstly progress through each process trajectory and compare effects with traceable satisfaction goals in the goal model. Effects are compared to the desireability and temporal ordering of effects in normative goals and descriptive satisfaction links are established as we progress through the trajectory.

We then analyze and classify each trajectory as either *normal* or *exceptional*. A normal trajectory in relation to the goal model leads to the satisfaction of all normative goals. An exceptional trajectory, as described in the previous section, satisfies a limited number of normative satisfaction goals.

In order for a satisfaction relationship to exist between a goal model and an associated process model, there must be at least one normal trajectory. That is, the process model must support at least one valid means by which to satisfy the required normative goals of the process.

Finally, we analyze the outcome of the satisfaction process, identifying whether the process supports the achievement normative satisfaction goals and classify the satisfaction relationship between the process and the associated goal model as either *strong*, *weak*, or *unsatisfied*. A *strong* satisfaction relationship is determined if all possible trajectories are ‘normal’ (i.e. satisfy all associated goals). On the other hand, a *weak* satisfaction relationship is said to exist when there is at least one ‘exceptional’ trajectory and one ‘normal’ trajectory. This classification, delineating between weak and strong satisfaction, can be important when evaluating the competency of the process in recovering from exceptional situations that may arise during enactment. An *unsatisfied* satisfaction relationship is the result of there not being a single ‘normal’ trajectory decomposed from the process. This classification requires that changes are made to either the process and/or goal model to establish a weak or strong satisfaction relationship.

## 6 Example

We apply the GoalBPM to a single case within a Transport Organization for illustration. GoalBPM is specifically applied to a core operational 'Package Sorting' process within the organization (see Figure 2). Furthermore, we introduce change to the goal model, and identify inconsistencies with the current business process that need to be addressed to maintain the satisfaction relationship.

### 6.1 Current Business Context and Process

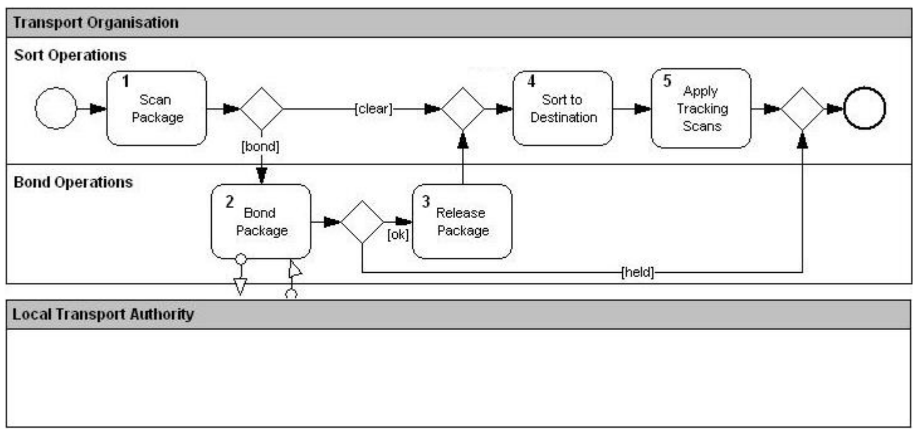


**Fig. 2.** The 'Package Sorting' Process of the Transport Organization

The organisation has recently chosen to undergo redesign to maintain effectiveness specifically in relation to increasing *regulatory requirements* including the screening of packages. This has impacted the internal *package sorting* operations of the organisation by introducing the requirement for every package to be screened by a representative of the authority upon arrival and prior to it leaving the sort facility for delivery or routing.

The operational objective the 'Package Sorting' process aims to achieve is the prompt routing of *packages*, upon arrival to a *sorting facility*, to their respective destinations. There are three primary participants in the process whose objectives must be met: *Transport Organization* - whose concern is the efficient routing of packages to their destinations by assigning responsibility to internal sort operations as well as bond operations whose role is to liaise with transport authorities for prompt delivery clearance; *Customers* - whose concern is prompt delivery and package traceability; and *Transport Authorities* - whose concern is with maintaining a high level of integrity in regards to border control through package screening. These requirements are represented on the goal model in Figure 3, with definitions supplied in Figure 4.

The ‘Package Sorting’ process in Figure 2, represents the current ‘as-is’ coordination of activities and interactions aimed toward achieving traceable goals. We apply GoalBPM to the process and goal model in order to evaluate the current state of the satisfaction relationship before introducing the aforementioned changes to the goal model.

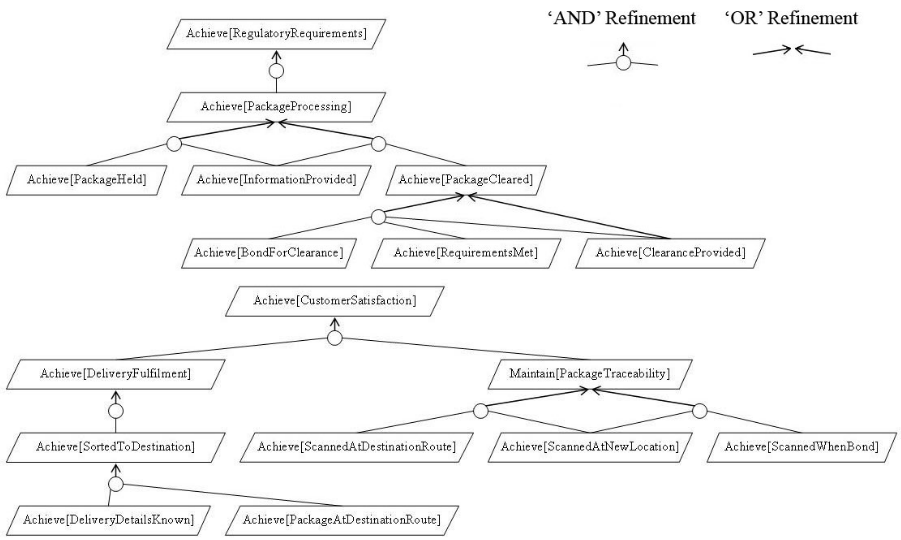


**Fig. 3.** Goals Traceable to the ‘Package Sorting’ Process

## 6.2 Applying GoalBPM

We apply the proposed GoalBPM to prove the satisfaction relationship between the goal model and the process model. We specify effects as they would be defined formally due to space limitations (e.g. *Arrives(p, sf)* is equivalent to *PackageArrivesAtTheSortFacility*, as can be inferred from the goal definitions in Figure 4).

**Analyzing Traceability** We firstly declare boundaries to help guide an evaluation of the traceability relation by identifying the required *pre* and possible *post* conditions for the ‘Package Sorting’ process. *Pre* conditions include: Arrives(p, sf) AND Provided(p.DeliveryDetails, ta). *Post* conditions include: Sorted(p, p.Destination) OR Held(p, ta).

The initial pass at goal traceability for the ‘Package Sorting’ process identified three high-level goals. Further analysis identifies the specific refinements that are required for satisfaction at some point during process enactment. These goals are declared in Figure 3 and defined in Figure 4.

| Goal Type | Goal Declaration | Concerned Objects | Antecedent | Consequent |
|---|---|---|---|---|
| Achieve | PackageProcessing | Package, SortFacility, TransportAuthority | Arrives(p, sf) | <>Processed(p, ta) |
| Achieve | InformationProvided | Package, SortFacility, TransportAuthority | Arrives(p, sf) | <>Provided(p.DeliveryDetails, ta) |
| Achieve | PackageCleared | Package, TransportAuthority | Provided(p.DeliveryDetails, ta) | <>Cleared(p, ta) |
| Achieve | PackageHeld | Package, TransportAuthority | Provided(p.DeliveryDetails, ta) | <>Held(p, ta) |
| Achieve | BondForClearance | Package, SortFacility, TransportAuthority | Provided(p.DeliveryDetails, ta) | <>Bond(p, sf) *W* Passed(p, ta.Requirements) |
| Achieve | RequirementsMet | Package, SortFacility TransportAuthority | Bond(p, sf) | <>Passed(p, ta.Requirements) |
| Achieve | ClearanceProvided | Package, TransportAuthority | Provided(p.DeliveryDetails, ta) ^ Passed(p, ta.Requirements | <>Cleared(p, ta) |
| Achieve | DeliveryFulfillment | Package, SortFacility, Customer | Arrives(p, sf) | <>Delivered(p, c) |
| Achieve | SortedToDestination | Package, SortFacility | Arrives(p, sf) | <>Sorted(p, p.Destination) |
| Achieve | DeliveryDetailsKnown | Package, SortFacility | Arrives(p, sf) | <>Known(p.DeliveryDetails, sf) |
| Achieve | PackageAtDestinationRoute | Package, TransportAuthority | Known(p.DeliveryDetails, sf) | <>Sorted(p, p.Destination) |
| Achieve | ScannedAtNewLocation | Package, SortFacility | Arrives(p, sf) | <>Scanned(p, sf) |
| Achieve | ScannedAtDestinationRoute | Package, SortFacility | Sorted(p, p.Destination) | <>Scanned(p, sf) |
| Achieve | ScannedWhenBond | Package, SortFacility, TransportAuthority | Bond(p, sf) | <>Scanned(p, sf) |

**Fig. 4.** Definitions for Traceable 'Package Sorting' Goals

| # | Activity | Effect Annotation |
|---|---|---|
| | *Pre Conditions* | *Arrives(p, sf) AND Provided(p.DeliveryDetails, ta)* |
| | *Possible Effects* | *Passed(p, ta.Requirements) AND Cleared(p, ta)* |
| 1 | Scan Package | Known(p.DeliveryDetails, sf) AND Scanned(p, sf) |
| 2 | Bond Package | Bond(p, ta) AND Scanned(p, sf) |
| | | Bond(p, ta) AND Scanned(p, sf) AND Held(p, ta) |
| 3 | Release Package | Passed(p, ta.Requirements) AND Cleared(p, ta) |
| 4 | Sort to Destination | Sorted(p, p.Destination) |
| 5 | Apply Tracking Scans | Scanned(p, sf) |

| T# | A# | Cumulative Effect Assessment |
|---|---|---|
| 1 | 0 | Arrives(p, sf) AND Provided(p.DeliveryDetails, ta) AND Passed(p, ta.Requirements) AND Cleared(p, ta) |
| | 1 | Known(p.DeliveryDetails, sf) AND Scanned(p, sf) |
| | 4 | Sorted(p, p.Destination) |
| | 5 | Scanned(p, sf) |
| 2 | 0 | Arrives(p, sf) AND Provided(p.DeliveryDetails, ta) |
| | 1 | Known(p.DeliveryDetails, sf) AND Scanned(p, sf) |
| | 2 | Bond(p, ta) AND Scanned(p, sf) AND Held(p, ta) |
| 3 | 0 | Arrives(p, sf) AND Provided(p.DeliveryDetails, ta) |
| | 1 | Known(p.DeliveryDetails, sf) AND Scanned(p, sf) |
| | 2 | Bond(p, ta) AND Scanned(p, sf) |
| | 3 | Passed(p, ta.Requirements) AND Cleared(p, ta) |
| | 4 | Sorted(p, p.Destination) |
| | 5 | Scanned(p, sf) |

| T# | A# | Goal Satisfaction |
|---|---|---|
| 1 | Pre | InformationProvided |
| | | PackageCleared |
| | | ClearanceProvided |
| | 1 | DeliveryDetailsKnown |
| | | ScannedAtNewLocation |
| | 4 | PackageAtDestinationRoute |
| | | SortedToDestination |
| | 5 | ScannedAtDestinationRoute |
| 2 | Pre | InformationProvided |
| | 1 | DeliveryDetailsKnown |
| | | ScannedAtNewLocation |
| | 2 | PackageHeld |
| | | ScannedWhenBond |
| 3 | Pre | InformationProvided |
| | 1 | DeliveryDetailsKnown |
| | | ScannedAtNewLocation |
| | 2 | ScannedWhenBond |
| | 3 | PackageCleared |
| | | RequirementsMet |
| | | ClearanceProvided |
| | 4 | SortedToDestination |
| | | PackageAtDestinationRoute |
| | 5 | ScannedAtDestinationRoute |

**Fig. 5.** Tabulated Effect Annotation, Trajectory Decomposition, and Goal Satisfaction

**Effect Annotation** Firstly, we annotate the model with effects to identify the achievable (and alternative) outcomes of activities in the current process. We also include the pre-conditions themselves, and any other relevant/influential effects that may be caused by other processes that have a direct impact on process decisions and coordination. These annotations are listed in Figure 5.

Process initiation is governed by two conditions: the *arrival of packages to the sort facility* and the *provision of package information to transport authorities*. It is also identified that the prior provision of information to authorities may allow for the rapid *clearance of packages for delivery* prior to the sorting process initiating. This may occur if the requirements of the authority can be identified as being met. These effects are also added to the list of relevant/influential effects that may have occurred prior to process initiation.

Each activity is then analyzed and annotated with normal and exceptional effects. *Scanning a package* results in the *delivery details being known*. The package may be *bonded for clearance* with another *scan* being applied, and alternatively *held* if the transport authority requests. The latter is an exceptional effect that occurs due to the package meeting some characteristics that require it to be given to the authorities that then take sole control of the package from that point on. The outcome of releasing a package is the *passing of transport authority requirements* and ultimately *clearance*. The sorting activity then results in the *package being sorted to its destination*. The final *scanning* activity results in another update of package location.

**Trajectory Decomposition** We now decompose trajectories. An analysis of the conditions influencing the choice of paths at decision gateways also guides the choice of prior and/or alternative effects when accumulating them for any given trajectory. We identify that that there are three trajectories represented in the process model. These are listed in Figure 5 with the actions that compose the trajectory, and the effects chosen from associated annotations.

The first trajectory decomposed from the model represents the *prior clearance of a package*, resulting in the eventual *sorting of the package* to its destination. The second trajectory results in packages requiring *bonding prior to clearance*, however the exceptional alternative is selected based on the adjacent decision gateway, and the package is *held by the transport authority*. The final trajectory is categorized by *requiring the package to be bonded* however, the package is eventually *cleared by passing the authorities requirements*, as well as being *sorted* to, and *scanned at its destination*.

**Satisfaction Analysis** We iterate through each trajectory, accumulate effects, and correlate accumulations with the desired effects and their temporal ordering on the goal model. Goals must be satisfied by firstly achievement of their pre conditions (i.e. the antecedent) with the subsequent achievement of the consequent in conformance to the temporal pattern chosen. Goal definitions are listed in Figure 4, with the results of the satisfaction process listed in Figure 5.

The satisfaction process identifies two normal trajectories and one exceptional trajectory representing a *weak* satisfaction relationship between the goal model and the process model. The achievement of the operational objective to sort packages to their destinations occurs in trajectories (1) and (3), however it is not achieved in (2) due to an exceptional result occurring when the package is bonded (i.e. it is held).

### 6.3 Changes to the Goal Model

We now introduce the newly acquired regulatory requirements for package screening that were mentioned previously. These requirements are added to the required goals for satisfaction by the process.

**Goal Alterations** Required alterations to the goal model, whether they be the addition, removal or modification of goals, ultimately result in modification to the *desirability* and/or *temporal ordering* of effects. Take for example, the newly acquired requirement for package screening, as represented on the adjusted portion of the goal model in Figure 6. This has introduced a newly desired effect requiring *all packages be screened by the transport authority once they arrive at a sort facility*, or formally:

$\forall$p: Package, sf: SortFacility, ta: TransportAuthority
Arrives(p, sf) $\Rightarrow \diamond$Screened(p, ta)

Additions of new effects to goal models frequently impact other goals and subsequently their refinements. As noted on Figure 6, the desired *Screened(p, ta)* is also added to the pre conditions for *PackageHeld*, as well as the *PackageCleared* goal and its refinements.

**Process Implications and Evolution** Any alterations to the goal model will proportionately affect the desired *achievement* or *coordination* of effects within the business processes that are assigned their operationalization. We re-evaluate the satisfaction relationship between the 'Package Sorting' process and its goals, and apply some informal analysis to identify specific changes required at the process level.

Upon evaluation of the satisfaction relationship, it is identified that previously normal trajectories (i.e. (1) and (3)), are now also exceptional due to their inability to satisfy regulatory requirements. This is consistent with the modifications to the goal model.

Upon further analysis of unsatisfied goals, we deduce that the *Screened(p, ta)* effect is required at some time after package arrival and prior to package clearance including bonding, as well as package holding. A decision is made as to the addition of the desired screening outcome within the process as an activity shown in Figure 7.

The process model is then re-evaluated. The included activity (6) is annotated with a normal effect that realizes *Screened(p, ta)*, and the exceptional effect of

*Held(p, ta)* is also identified as possibly resulting from the activity. A decision gateway is applied adjacent to the screening activity to evaluate the outcome of the activity, and redirect process flow. Upon trajectory decomposition a new exceptional trajectory is decomposed resulting from the prior addition of the decision gateway. Goal satisfaction is then re-evaluated. The changes to the process are successful at resolving the unsatisfied relationship by achieving a weak satisfaction relationship with two normal and two exceptional trajectories.

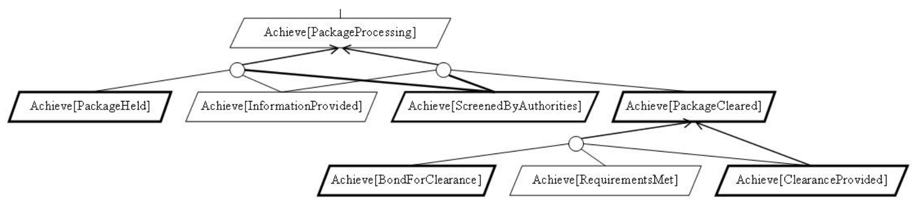


**Fig. 6.** Goal Model Additions for Screening Requirements

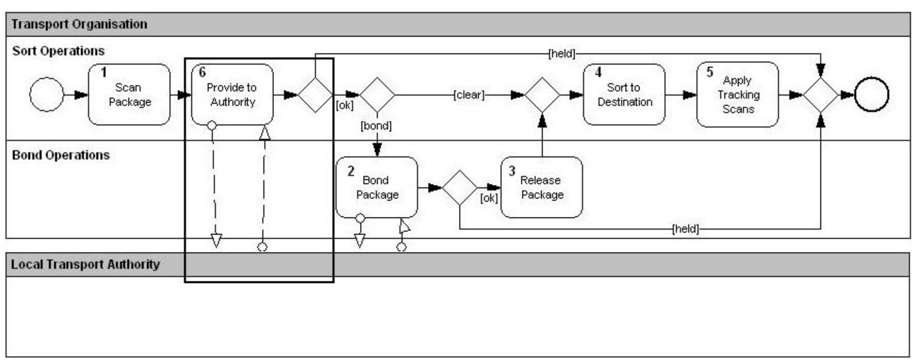


**Fig. 7.** Process Evolution to Achieve Goal Satisfaction

## 7 Conclusion

We have proposed the GoalBPM methodology that can be used to identify the satisfaction of a process model against a goal model. The example we presented, provides a brief and informal overview. There are many possible benefits in applying GoalBPM to current business process design and analysis. This includes the initial intentional design of business processes that satisfy a deliberate specification of goals. Changes to the business process model may then be made and tested against the specification of the goals they wish to satisfy in the goal model. Changes may also be made to the goal model and tested against the current business process model to identify behaviors that are invalidated. Invalid behavior may be explicitly defined, supporting further redesign to align the processes changed against organizational goals. In order to progress from the current state, the need for formalism, tool support and testing against a large, non-trivial business case is required. We are actively pursuing these requirements, which we hope will increase our understanding of the realizability, workability and viability of GoalBPM for its active use.

## References


1. Smith, H., Fingar, P.: Business Process Management: The Third Wave. Meghan-Kiffer Press, Tampa, FL (2003)
2. Dumas, M., van der Aalst, W.M., ter Hofstede, A.H.: Process-Aware Information Systems: Bridging People and Software Through Process Technology. Wiley-Interscience (2005)
3. McGoveran, D.: The benefits of a bpms. Technical report, Alternative Technologies, Felton, California, USA (2002)
4. van der Aalst, W., ter Hofstede, A., Weske, M.: Business process management: A survey. In: BPM'03 - International Conference on Business Process Manag ement, Berlin, Springer-Verlag, Lecture Notes in Computer Science (2003) 1–12
5. Youngblood, M.D.: Winning cultures for the new economy. Strategy and Leadership **28**(6) (2000) 4–9
6. Kavakli, E.: Modelling organizational goals: Analysis of current methods. In: Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, CY (2004) 1339–1343
7. Pyke, J., Whitehead, R.: Do better maths lead to better business processes? Business Process Trends, http://www.bptrends.com (2004)
8. Wynn, D., Eckert, C., Clarkson, P.J.: Planning business processes in product development organisation s. In: REBPS'03 - Workshop on Requirements Engineering for Busines s Process Support, Klagenfurt/Veldern, Austria (2003)
9. Watkins, R., Neal, M.: Why and how of requirements tracing. IEEE Software **11**(4) (1994) 104–106
10. van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: RE'01 - International Joint Conference on Requirements Engi neering, Toronto, IEEE (2001) 249–263
11. White, S.: Business Process Modeling Notation (BPMN), Version 1.0. Business Process Management Initiative (BPMI.org). 1.0 edn. (2004)
12. Letier, E., van Lamsweerde, A.: Deriving operational software. In: FSEí10 - 10th ACM SIGSOFT Symp. on the Foundations of S oftware Engineering. (2002)
13. Letier, E.: Reasoning about Agents in Goal-Oriented Requirements Engineerin g. PhD thesis, Universite Catholique de Louvain, Louvain, Belgium (2001)
14. van Lamsweerde, A., Letier, E.: Handling obstacles in goal-oriented requirements engineering. IEEE Transactions on Software Engineering **26**(10) (2000) 978–1005
15. Bider, I., Johannesson, P.: Tutorial on: Modeling dynamics of business processes – key for building next generation of business information systems. in: The 21st international con-ference on conceptual modeling (er2002), tampere, fl, october 7-11, 2002. In: 21st International Conference on Conceptual Modeling (ER2002), Tampere, FL (2002)

16. Khomyakov, M., Bider, I.: Achieving workflow flexibility through taming the chaos. In: OOIS'00 - 6th International Conference on Object Oriented I nformation Systems, Springer-Verlag, Berlin (2000) 85–92
17. Andersson, T., Andersson-Ceder, A., Bider, I.: State flow as a way of analysing business processes - case stud ies. Logistics Information Management **15**(1) (2002) 34–45