Combining Intention-Oriented and State-Based Process Modeling

Pnina Soffer[1] and Colette Rolland[2]

[1]MIS Department, Faculty of Social Science, Haifa University,
Carmel Mountain, Haifa 35901, Israel
spnina@is.haifa.ac.il
[2]Universite Paris1 Panthéon Sorbonne CRI, 90 rue de Tolbiac,
75013 Paris, France
Colette.rolland@univ-paris1.fr

Abstract. Business process modeling and design has gained importance in recent years. Consequently, a large number of modeling languages have emerged. Many of them lack formality, whereas some others support the verification of the designed process. Most of existing modeling languages adopt an operational view focusing on how the process is performed. By contrast, others follow the human intention of achieving a goal as the force that drives the process, and concentrate on what the process must do, i.e. on its rationale. The aim of this paper is to combine intention-oriented modeling with formal state-based modeling and achieve their synergy, benefiting from the advantages of both. We use the Map formalism as an example of the former and the Generic Process Model (GPM) as an example of the latter. The paper proposes a procedure for converting a Map into GPM concepts, illustrates it with the SAP Material Management Module and shows the benefits resulting from it.

1 Introduction

Conceptual modeling is aimed at representing the real world for purposes such as understanding, communicating, and reasoning in the process of information systems analysis and design. An important area that emerged in recent years is business process modeling, whose main focus is capturing behavioral aspects of the world, but it also relates to other aspects. Various types of process modeling languages and formalisms have emerged, supporting a variety of purposes. The existing formalisms can be roughly classified according to their orientation to activity-sequence oriented languages (e.g., UML Activity Diagram), agent-oriented languages (e.g., Role-Activity Diagram [7]), state-based languages (e.g., UML statecharts), and intention-oriented languages (e.g., Map [9]). Many of these languages lack formality, and serve as a graphical tool assisting in the creative task of process and IS design. The lack of formality makes the analysis and verification of the designed processes a difficult task.

The concept of goal is central in business process modeling and design. It is included in many definitions of business processes (e.g., "a business process is a set of partially ordered activities aimed at reaching a goal" [6]). However, most process modeling languages do not employ a goal construct as an integral part of the model. This is sometimes justified by viewing these models as an "internal" view of a process, focusing on *how* the process is performed and externalizing *what* the process is intended to accomplish in the goal [5].

In contrast, intention-oriented process modeling focuses on *what* the process is intended to achieve, thus providing the rationale of the process, i.e. *why* the process is performed. Intention-oriented process modeling follows the human intention of achieving a goal as a force that drives the process. As a consequence, goals to be accomplished are explicitly represented in the process model together with the different alternative ways for achieving them, thus facilitating the selection of the appropriate alternative for achieving the goal.

Process goals are also present in some state-based modeling formalisms (e.g., [2]). However, as opposed to intention oriented goals, in state-based modeling a goal stands for a state or a set of states on which the process terminates. State-based modeling captures a process as a flow of states, leading to the goal state. This representation of a process takes a structural view rather than an intentional view, and can be formal enough to provide a basis for analyzing the properties of a process model and its validity [13].

The main difference between the goal concept in intention-oriented modeling and state-based modeling is that in the former, while aiming at representing the human intention, goals are not formally defined and may bear a rather vague meaning. In the latter, in contrast, goals are formally defined, but are not directly related to the human intention.

The aim of this paper is to combine intention-oriented modeling with state-based modeling and achieve their synergy, benefiting from the advantages of both. The intention-oriented modeling notation we use is the Map formalism [9, 11], and the state-based modeling notation is the Generic Process Model (GPM) [12, 13]. The Map notation is intuitive and easy to apply and understand. It is particularly suitable for representing unstructured processes, whose sequence of activities may vary in different situations, or processes including variability (e.g. product lines, ERP or adaptable processes), whose sequence of activities is selected at run time depending on the situation at hand. However, Maps are not formally defined, hence there is no structured procedure for analyzing a Map for deficiencies and invalidity. Furthermore, while the map concepts are intuitively understood, there is no precise definition to their semantics. We suggest a procedure for converting a Map to the state-based concepts of GPM, and use the formality and precision gained in order to achieve a better understanding of the concepts underlying the map. In particular, this understanding provides insights to the essence of acting on an intention. The result is an intention-oriented model which is formally defined and can be analyzed for completeness and validity.

The remainder of the paper is organized as follows: Section 2 provides an overview of both modeling formalisms, Map and GPM; Section 3 interprets Map concepts in GPM terms; Section 4 presents a procedure for transforming Map representation into GPM model, illustrates it by an example, and demonstrates how a model can be analyzed and improved by applying this transformation; conclusions are given in Section 5.

2 The Map and GPM formalisms

This section provides an overview of both modeling formalisms, the Map and GPM.

2.1 An overview of Map

In this section we introduce the concept of a map and illustrate it with the *Material Management map (MM map)*, based on the information provided in [1] regarding the SAP R/3 Materials Management module.

The *Map* representation system allows to represent a process model expressed in intentional terms. It provides a representation mechanism based on a non-deterministic ordering of *intentions* and *strategies*.

A *map* is a labeled directed graph (Figure 1) with *intentions* as nodes and *strategies* as edges. An edge enters a node if its strategy can be used to achieve the intention of the node. Since there can be multiple edges entering a node, the map is capable of representing many strategies that can be used for achieving an intention.

An *intention* represents a goal that can be achieved by the performance of a process. For example, the MM map in Figure 1 has *Purchase Material* and *Monitor Stock* as intentions. Furthermore, each map has two special intentions, *Start* and *Stop*, to respectively start and end the process.

A *strategy* is an approach, a manner to achieve an intention. In Figure 1 *Manual strategy* is a manner to manually generate an order to *Purchase Material.*

A *section* is a key element of a map. It is a triplet as for instance <*Start, Purchase Material, Planning Strategy*> which couples a source intention (*Start*) to a target intention (*Purchase Material*) through a strategy (*Planning strategy*) and represents a way to achieve the target intention *Purchase Material* from the source intention *Start* following the *Planning Strategy*. Each section of the map captures a specific manner in which the process associated with this goal can be performed. A section may be recursive when its source and target intentions are the same. In Figure 1, the section <*Purchase Material, Purchase Material, reminder strategy*> is recursive.

Sections of a map are *connected* to one another. This occurs:

(a) When a given goal can be achieved using different strategies. This is represented in the map by several sections between a pair of intentions. The topology corresponding to the case where several strategies can be selected is called a *multi-thread*. In Figure 1 the multi-thread between *Start* and *Purchase Material* represents the two ways in which the *Purchase Material* intention can be achieved (manually and by planning). When the strategies are mutually exclusive, sections are said to constitute a *bundle*, specified by a dotted line and refined in a separate map. In Figure 1, *Planning strategy* is a bundle composed of two exclusive strategies to achieve the Purchase Material intention, namely *By reorder point planning* and *by forecast based planning*, as shown in Figure 2. Only one of these can be selected each time the *Planning strategy* is taken.

(b) When an intention can be achieved by several combinations of strategies. This is represented in the map by a pair of intentions connected by several sequences of sections. Such a topology is called a *multi-path*. In Figure 1 there are five paths leading from *Start* to *Monitor Stock*.

In general, a map is a multi-path from *Start* to *Stop* and contains bundles and multi-threads. Figure 1 contains several paths from *Start* to *Stop* to handle the "normal cases" and complete the process (i.e. to achieve the *Stop*) through the *Purchase Material* and the *Monitor Stock* intentions. This map also allows exceptional cases as, for instance, with the path that directly allows to *Monitor Stock* following the *By Bill for Expenses* strategy.

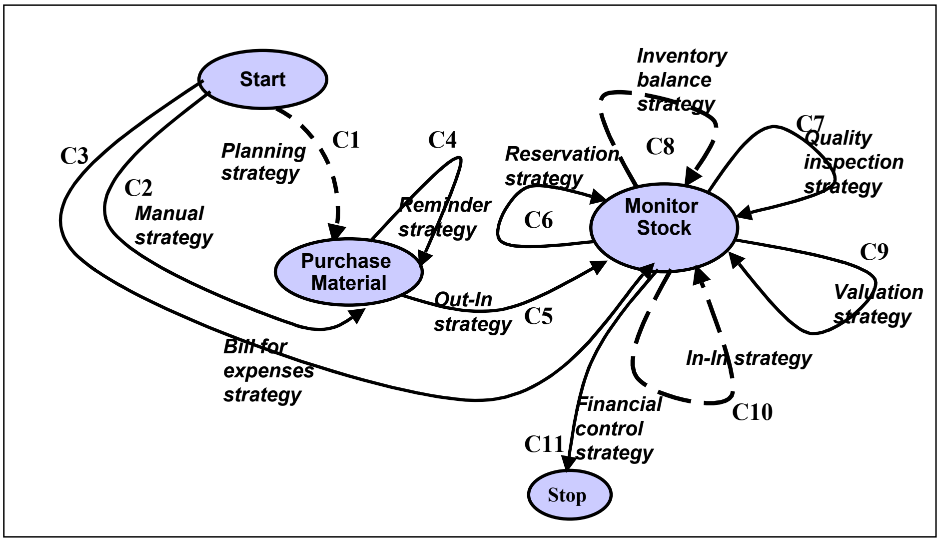

Figure 1. The material management map

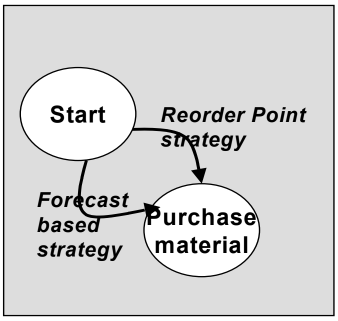

Figure 2. The Planning strategy bundle

Finally, a *section* of a map can be *refined as another map*. This happens when it is possible to view the section as having its own intentions and associated strategies. The entire refined map then represents the section.

As a consequence of its intentional orientation, a map does not represent a flow of tasks. Rather, it presents a non deterministic ordering of intention / strategy selections to accomplish the main process intention. Besides, given the multi-path and multi-thread topologies, a map is able to present a global perspective of the diverse ways of achievement the process intention. The map of Figure 1 for example, shows 25 paths from *Start* to *Stop*, 5 following the *Bill for Expenses strategy*, 10 following the *Planning Strategy*, and 10 following the *Manual strategy.* All these paths allow to achieve the main process intention namely*, Satisfy Material Need Efficiently*.

The Map was selected as the intention-oriented model to be formalized in this paper because, unlike other intention-oriented models (e.g., i* [15]), it captures the flow of a process and establishes a direct relation between a goal and the actions that can be taken in order to achieve it. As well, maps employ a small set of constructs (as compared to i*) and therefore allow us to concentrate on the concept of intention and the way it drives a process.

2.2 An overview of GPM

GPM is based on Bunge's ontology [3, 4], as adapted for information systems modeling (e.g., [8, 14]), for conceptual modeling, and for modeling business process concepts.

According to the ontological framework, the world is made of *things* that possess *properties*. Properties can be *intrinsic* (e.g. height) to things or *mutual* to several things (e.g. a person works for a company). Things can compose to form a *composite* thing that has *emergent* properties, namely, properties not possessed by the individuals composing it. Properties (intrinsic or mutual) are perceived by humans in terms of *attributes*, which can be represented as functions on time. The *state* of a thing is the set of values of all its attribute functions (also termed state variables). When properties of things change, these changes are manifested as state changes or *events*. State changes can happen either due to internal transformations in things (self action of a thing) or due to *interactions* among things. Not all states are possible, and not all state changes can occur. The rules governing possible states and state changes are termed *state laws* and *transition laws*, respectively. States can be classified as being *stable* or *unstable*, where an unstable state is a state that must change by law, and a stable state is a state that can only change as a result of an action of something external to the thing or the domain.

A *domain* is a part of the world of which we wish to model changes, and represents the scope of our control. A domain is a set of things and their interactions, and is represented by a set of state variables, which stand for the intrinsic and mutual properties of these things, including emergent properties of the domain itself. A *sub-domain* is a part of the domain, represented by a subset of the domain state variables. A sub-domain may be in a stable state while the entire domain is in an unstable state, meaning that a different part of the domain is currently subject to changes.

A *process* is a sequence of unstable states, transforming by law until a stable state is reached. A process is defined over a domain, which sets the boundaries of what is in a stable or an unstable state. Events that occur outside the domain are *external events* and they can activate the domain when it is in a stable state.

A process model in GPM is a quadruple <S, L, I, G>, where S is a set of states representing the domain of the process; L is the law, specified as mapping between subsets of states; I is a subset of unstable states, which are the initial states of the process after a triggering external event has occurred; G is a subset of stable states, which are the goal of the process. Subsets of states are specified by conditions defined over criterion functions in the state variables of the domain. Hence, a process starts when a certain condition on the state of the domain holds, and ends when its goal is reached, i.e., when another condition specified on the state of the domain holds. For example, the initial set of a production process can be specified as {s| Production Order Status = "Released" AND Materials = "Available" AND Resources = "Available"}, which is a set of unstable states (that became unstable by the release of the production order). The goal of this production process can be specified as {s| Production Order Status = "Completed" AND Quality = "Approved"}, which is a set of stable states. The states in the goal set may differ from each other in the values of state variables such as production time and cost. Nevertheless, they all meet the condition specified. The criterion function defines the set of state variables that are relevant for determining that the process has reached its goal.

3 Interpreting Map in GPM terms

In this section we interpret the concepts of Map in GPM terms and establish a set of concepts which is common to both formalisms.

3.1 Basic concepts

A Map is specified as a set of intentions or goals to be achieved, and strategies for achieving them. The goals can be interpreted as sets of desired states, on which the strategies terminate. However, taking an action (strategy) aimed at reaching a goal does not necessarily end in attaining it. Some goals may require a number of actions to be performed before being achieved. In other cases an action may or may not achieve the goal, and, based on the result accomplished, that action may be repeated or another action may be taken. Still, the Map notation specifies every strategy as leading to an intention, even if it is not able to immediately achieve the goal. In fact, the intention a strategy leads to specifies *why* this strategy is taken rather than the goal it actually achieves. Hence, transforming this into state-based concepts, we may view an intention to which a strategy leads as having a "core", which is the goal to be attained, and a broader set of states in which some action towards attaining the goal has already been performed. Taking an action (strategy) aimed at reaching the goal, does not necessarily end in attaining the goal, but should reach a state which is closer to that goal than the one prior to the action. All the strategies that lead to a given intention must at least end in a state where some action towards attaining the goal has been done.

Hence, defining an intention in state-based terms should include two parts: a basic subset of states indicating that some action has been performed, and the goal set.

Formally expressed:

An *Intention* I is specified as $<B_I, G_I>$, where B_I is the intention *Basic* subset of states and G_I is the intention *Goal*.

Being "in" an intention I means being in a state $s \in B_I$. The Goal set is, naturally, a subset of the basic set of the intention, $G_I \subseteq B_I$. Based on the GPM notation, all the subsets of states are specified in terms of conditions over criterion functions.
For example, B_I of the intention *Purchase Material* in Figure 1 is the set of states where a purchase order was issued, and G_I is the set of states where the goods arrived from the supplier.
A *strategy* is an action by which an intention can be achieved.
A *section* in a Map is comprised of a source intention, a target intention, and a strategy leading from the source to the target. In GPM terms, a section is a mapping between subsets of states, hence it specifies the law. When going from one intention to another, a strategy may be selected based on (a) preferences and success expectations, or (b) the current situation. The latter case means that a strategy does not necessarily start on any state $s \in B_I$. Rather, it can start on a subset of the states in the Basic subset of its source intention. Similarly, it leads to a state belonging to a subset of its target intention Basic set, which is a result of the specific actions of the strategy. In Figure 1 the *Bill for Expenses* strategy will start on a subset of "emergency" states, and end on a subset of states where goods arrive and expenses are billed for. Two or more strategies that share a common initial subset of states are a bundle. It is therefore clear that a section is not only defined by the Basic set of its source and target intentions, but by more specific subsets of initial and final states. Formally expressed:
A section S from intention I to intention J is a mapping between an initial subset of states I_S and a final subset of states F_S, such that:

(a) $I_S \subseteq B_I$

(b) $F_S \subseteq B_J$

Going from intention I to intention J means trying to achieve the goal of J. Even if this goal is not achieved, the strategy taken should result in a state which is "closer" to that goal than the one prior to the action taken. We would like to be able to state that $|G_J - F_S| < |G_J - I_S|$. This is obvious when the section includes two different intentions, leading from intention I to J, since the desired goal $G_J \subseteq B_J$, on which the section ends.
However, for recursive sections, whose source and target intentions are the same one, the notion of distance between subsets of states should be examined.
Assume the goal criterion function relates to a single state variable (i.e., it has a single dimension). We shall also assume that this state variable exists in a domain of values where the operators >, <, = hold. This means that there is some kind of ordinality in the values that may be attained by state variables. This ordinality may be numerical, preference-based, or a result of procedural sequence. Then, moving along this dimension, the distance from the goal changes, and one can clearly identify that a strategy ends on a state which is closer to the goal than the state before. However, a criterion function may relate to a number of state variables (i.e., be multi-dimensional). Furthermore, these dimensions may have trade-off relations among them. Computing a precise distance from the goal in such situations may involve weighing techniques, thus the computed distance would depend on the weight assigned to each dimension. Nevertheless, changes in the distance with respect to each dimension separately can still be straightforward and easily perceived.
As an example, assume an intention of improving a production process. The improvement may be in terms of cost, time, and quality, which have trade-off

relations among them. The improvement intention can be achieved through different strategies: cost-reduction strategy, quality-improvement strategy, time-reduction strategy, and so on. However, the cost-reduction strategy may inversely affect the product quality, the quality-improvement strategy may increase the cost, the time-reduction strategy may decrease the cost but damage the quality, etc. Yet, they are all valid strategies.

For any practical purpose, we may assume the process designer can clearly identify and evaluate whether a strategy "contributes" to achieving a goal along each dimension, and whether a final subset of states is "closer" to a goal set. For such purposes, a move along at least one dimension will be sufficient for determining that such contribution is made, and we can expect different strategies to contribute along different dimensions. Furthermore, a Map is a model that specifies possible and alternative paths (combinations of strategies), thus facilitating the selection of an appropriate strategy at run time, when the process is executed. Such selection may take into account trade-offs among dimensions and assign appropriate weights in the course of the decision making. However, this decision and its factors are situation-dependent, and not a part of the process modeling and design phase.

In summary, the computation of an absolute distance between subsets of states is situation dependent. The strategies in a map should clearly reduce the distance from the goal set along one dimension, thus establish a possible path to be taken, where the actual decision whether to take this path can be made in run time.

3.2 Section classification

To gain more understanding about the nature of processes modeled by Maps, we shall now elaborate on types of sections, and differentiate them to classes based on their behavior. A section may either be recursive or non-recursive. Its initial subset of states (I_S) is a subset of the basic set of its source intention, and its final subset of states (F_S) is a subset of the basic set of its target intention. As discussed above, the final subset of states should be defined so that the section ends in a state whose distance from the goal is smaller than the initial distance.

Figure 3 outlines possible cases of sections, their initial and final subsets with respect to their source and target intentions.

Cases 1 and 2 are cases where the strategy leads to a state closer to the goal than the initial state, but not to the goal itself. In case 1 the strategy leads from intention I to intention J. Its initial set is a subset of the goal of I, as this goal must have been achieved before proceeding to the next intention. Reaching intention J means reaching a state where some action towards attaining the goal has been performed, but has not actually achieved the goal. It might be that the goal requires a number of actions to be performed, in which case a recursive strategy can be taken, or that the goal achievement depends on an external event yet to occur. For example, one may have an intention of calling a meeting to discuss a certain issue. A strategy of preparing a presentation and materials for discussion will bring to a state which is closer to the goal, but the goal itself will not be achieved until the other participants will arrive.

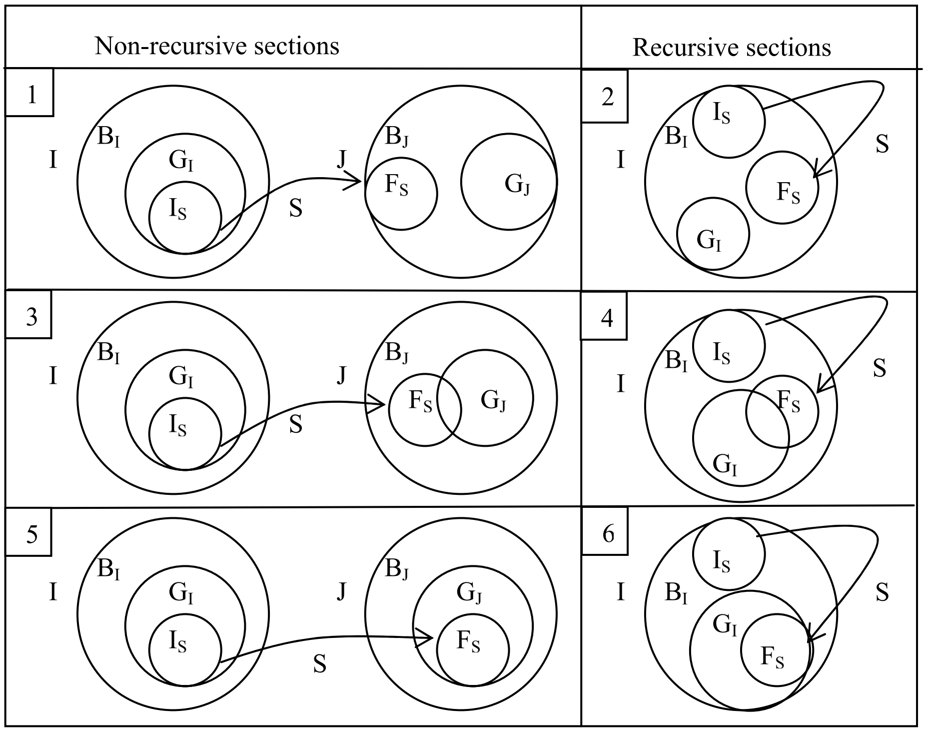

Figure 3. Possible cases of Initial and Final subsets of sections

Case 2 is of a recursive strategy, going between two subsets of the same intention, getting closer to the goal, but not reaching it. A recursive strategy can be taken only after some action towards the goal has already been performed, so the current (initial) state belongs to the basic set of the intention. For example, after preparing the presentation and material for the meeting discussed earlier, a recursive strategy of this type would be to send reminders to all the participants. This strategy is a move in the direction of the goal (the meeting), but cannot reach the goal itself.

Cases 3 and 4 are cases where the strategy may lead to the goal and may not, as its final set of states partly overlaps the goal set. These are cases where only after an action was taken one can see if the goal has been achieved or if another (recursive) strategy should be taken. For example, when a mechanic tries to fix a car, he cannot be certain that the action he is taking will solve the problem indeed. If the problem is not solved, then he uses the new information he gained to reassess the situation and decide what his next strategy will be. In other words, the strategies that belong to cases 3 and 4 have a potential of reaching the goal, but are not certain to do so. If the goal is not reached by them then a recursive strategy is still needed.

Cases 5 and 6 are cases where the strategy leads to the goal directly and with certainty. For example, producing an item is a strategy that leads to the goal of having the item available.

It may seem as if there is a seventh case, which is unique to intentions of maintaining a certain state that has already been reached. In such cases some recursive strategies are aimed at verifying that the desired state is not violated. As an example, consider the intention of keeping one's body in a good health, which has a recursive strategy of periodical physical examinations. It may seem that the initial state of such strategies is *in* the goal set. However, it is not certain to be so. Uncertainty makes the initial state of these strategies to be outside the goal set, and verification brings the final state back to being in the goal.

The above analysis leads to the following general results:

Result 1: Let S be a non-recursive section (I, J, s) then $I_S \subseteq G_I$.

This result is based on the assumption that a new intention will be sought only once the former one has been achieved. This is true in most cases. However, there are cases where an intention is temporal, and ceases to exist at a moment in time even if it is not achieved. For example, the intention of saving a drowning person will cease to exist when it is clear that the person cannot be saved anymore. Then the strategy leading to whatever the next intention is, will not start at the goal set of the current intention.

Result 2: Let S be a recursive section (I, I, s) then $I_S \cap G_I = \varnothing$.

This result is straightforward, since a recursive strategy is not needed and will not be performed if we are already at the goal set of the intention.

4 Representation transformation and process analysis

In this section we propose a procedure for transforming a Map to a GPM model, demonstrate it using the material management example, and discuss the insights that can be gained by this transformation.

4.1 The transformation procedure

Based on the concepts discussed and defined in Section 3, transforming a map to a GPM model would include the following steps:

1. Define each intention:
   - (a) The intention Goal G_I, in terms of conditions over criterion functions.
   - (b) The intention Basic Set B_I, in terms of conditions over criterion functions, specifying states where some action towards the goal has been performed. $G_I \subseteq B_I$.
   - (c) The Start and Stop intentions can be defined as sets of states without distinction between a Basic Set and a Goal set.
2. Define sections as the law: a mappings between subsets of states (I_S and F_S), applying the following:
   - (a) If the section is between two different intentions I and J, then:
     - i. The initial set $I_S \subseteq G_I$, where the conditions additional to the Goal condition specify the situation in which a certain strategy is to be taken.

ii. The final set $F_S \subseteq B_J$, where the conditions additional to the Basic Set condition specify the situation after the actions that were performed as part of the specific strategy.

(b) If the section includes a recursive strategy from an intention I to itself, then:

i. The initial set satisfies $I_S \subseteq B_I$ and $I_S \cap G_I = \emptyset$.

ii. The final set F_S should be closer to G_I at least along one dimension of the G_I criterion function.

(c) Bundles are addressed as sets of sections, whose I_S is mutual.

3. Repeat for each section refinement, placing I_S as the Start intention of the refined map and F_S as the Stop intention of the refined map.

To demonstrate the procedure, we shall use the material management example, and transform the map shown in Figure 1 to GPM representation. For simplicity, we shall not elaborate on the details of the bundles included in the map.

Step 1: intention definition

The states defining the intentions are specified and explained in Table 1.

The *Start* and *Stop* intentions specify states where nothing has been done and where the specific material handling is finished, respectively. Notice that the Goal conditions form subsets of the sets specified by the Basic conditions for the intentions of *Purchase Material* and *Monitor Stock*.

Table 1. Intention definition

Intention	Basic condition	Goal condition	Explanation
Start	(*Purchase Requisition*: not existing) AND (*Purchase Order*: not existing) AND (*Bill for Expenses*: not existing).		Start is when nothing has been done for purchasing
Purchase Material	*Purchase Order Status* ≤ “delivered”	*Sourced Goods*= “arrived”	Basic condition is that a purchase order exists. The goal is the arrival of material.
Monitor Stock	*Sourced Goods* = ”in stock”	*Stock Data*(attributes) = *Stocked Goods*(properties)	Basic condition is the existence of material in stock. Monitoring is keeping the data an accurate representation of reality.
Stop	(*Invoice* = “verified”) AND [(*Payment* = “authorized”) XOR (*Payment* = “blocked”)]		Stop is when payment of purchased goods is passed to the finances module.

Step 2: define sections as the law

The states defining the sections are specified and explained in Table 2. The Initial and final conditions of each section include the basic or goal conditions of the relevant intentions (depending on the type section type), and additional conditions. The sections are marked according to their mark in Figure 1. The explanations provide some additional assumptions made about the initial and final states of the sections.

Table 2. Section definition

Section	Initial condition	Final condition	Explanation
C1	(*Purchase Requisition*: not existing) AND (Purchase_ Order: not existing) AND (*Bill for Expenses*: not existing) AND (*Material Planning Type* = “automatic”)	(*Purchase Order Status* < “delivered”) AND *Requisition Status* = ”converted to order”	The planning strategy applies to materials whose planning type is defined as automatic. Requisitions are generated and converted to orders
C2	(*Purchase Requisition*: not existing) AND (*Purchase Order*: not existing) AND (*Bill for Expenses*: not existing) AND (*Material Planning Type* = “manual”)	(*Purchase Order Status* < “delivered”) AND *Requisition Status* = ”approved”	The manual strategy applies to materials whose planning type is defined as manual. Requisitions are generated and approved.
C3	(*Purchase Requisition*: not existing) AND (Purchase_ Order: not existing) AND (*Bill for Expenses*: not existing) AND (*Required Date – Current Date* < XX)	(*Sourced Goods* = “in stock”) AND (*Expenses Billing* = “registered”)	The bill for expense strategy applies when the material is needed suddenly and urgently, for a date which is less than XX days from the current
C4	(*Purchase Order Status* < “delivered”) AND (*Delivery Date* < *Current Date*)	(*Purchase Order Status* < “delivered”) AND (*Order History*: Reminder registered)	The reminder strategy is taken when the delivery date passed. It sends a reminder and registers it.
C5	(*Purchase Order Status* < “delivered”) AND (*Sourced Goods* = “arrived”)	(*Sourced Goods* = “in stock”) AND (*Purchase Order status* = “delivered”)	The out-in strategy is taken when goods arrive and puts them in stock.
C6	(*Sourced Goods* = “in stock”) AND (*Reservation Request*)	(*Sourced Goods* = “in stock”) AND (*Stock Status* = “reserved”)	Stock status records a reservation request for the attributes to match reality
C7	(*Sourced Goods* = “in stock”) AND (*Quality* = “not verified”)	(*Sourced Goods* = “in stock”) AND (*Quality* = “approved”)	The quality attribute reflects uncertainty until quality is verified and approved
C8	(*Sourced Goods* = “in stock”) AND (*Balance checking* = “needed”)	(*Stock Data*(attributes) = *Stocked Goods*(properties)) AND (*Stock Documents* = “generated”)	The balance strategy serves for verifying and documenting that the data attributes match reality
C9	(*Sourced Goods* = “in stock”) AND (*Material Value* = “not recorded”)	(*Sourced Goods* = “in stock”) AND (*Material Value* = “recorded”)	Recording the material value sets the value attribute to match reality
C10	(*Sourced Goods* = “in stock”) AND (*Movement Request*)	(*Sourced Goods* = “in stock”) AND (*Inventory Transaction* = “recorded”)	Recording a physical movement so location data matches reality
C11	(*Stock Data*(attributes) = *Stocked Goods*(properties)) AND (*Invoice* = “arrived”)	(*Invoice* = “verified”) AND [(*Payment* = “authorized”) XOR (*Payment* = “blocked”)]	Verifying the invoice and authorizing/blocking its payment

Note, that most of the recursive sections of the *Monitor Stock* intention are not assumed to lead with certainty to the goal (which is a full match between the stock data as registered and its real properties). This is because each strategy contributes to the accurate representation with respect to a specific issue only (e.g., quality, location, reservation, etc.). These strategies may get to the goal along one of its dimensions, while the other dimensions are not involved. The only strategy that leads to the goal with certainty is the *Inventory Balance* strategy, where the accuracy of the data is verified completely.
The example used here does not include section refinements, hence, step 3 of the procedure shall not be demonstrated. A refinement of the section <*Purchase Material, Monitor Stock, Out-In strategy*> is presented in [10]. Applying the procedure to this refinement, the *Start* intention would be specified by the condition (*Purchase Order Status* < “delivered”) AND (*Sourced Goods* = “arrived”) and the *Stop* intention would be specified by (*Sourced Goods* = “in stock”) AND (*Purchase Order Status* = “delivered”).

4.2 Process analysis

Transforming a Map representation to GPM representation enables identification of anomalies and deficiencies in the represented process. In particular, we may find indications for incompleteness of the specification and discontinuity of the process.

Deficiency of specification may be indicated by one of the following three cases:
(1) An intention in which the Basic Set includes states where no outgoing strategy is defined. Formally expressed: $B_I - \bigcup_{I_S \subseteq B_I} I_S \neq \phi$.

(2) An intention for which no strategy leads to the Goal set (i.e., all its ingoing strategies belong to cases 1 and 2 according to the classification of Figure 3). Formally: $F_s \cap G_I = \varnothing$ for every S leading to I.

(3) A section (except sections including the *Start* intention) for which no strategy leads to the initial set. Formally: $F_{S'} \cap I_S = \varnothing$ for every $S' \neq S$.

Case (1) is definitely a case of incompleteness, since it indicates the existence of states for which the law does not specify how to proceed. Assume the section <*Start*, *Purchase Material*, *Bill for expenses* > is not in figure1. Thus, due to the (*Bill for Expenses* : not existing) part of the basic condition of Start, this rule would help discovering the map is lacking a strategy.

Cases (2) and (3) may either indicate incompleteness or discontinuity.
In case (2): if the intention goal can only be achieved as a result of an external event then the process is non-continuous. Otherwise, a strategy for achieving the goal needs to be defined for the law to be complete.
In the material management example, the *Purchase Material* intention does not have any strategy that leads to its goal. The goal, which is the arrival of goods, can only be

obtained by an external event. As suggested in [10, 13], in such cases a reminder strategy is needed in order to make sure that the process will not be waiting for that external event indefinitely. It is also possible to specify an exception handling strategy, to be taken if the external event does not occur after the reminder strategy has been taken. The exception handling strategy will cancel the purchase order and terminate the process.
In case (3): if the initial set of a section can only be achieved as a result of an external event then the process is non-continuous. Otherwise, a strategy for reaching a state where the next step can begin needs to be defined for the law to be complete.
In the material management example the *Financial Control* strategy can only start when an invoice has arrived (external event). A reminder strategy is needed in order to make sure that this happens. However, there is no explicit intention to which such strategy should lead. This may indicate that payment for goods is an intention by itself, which should be separated from the *Stop* intention.
The result of the analysis is presented in Figure 4.

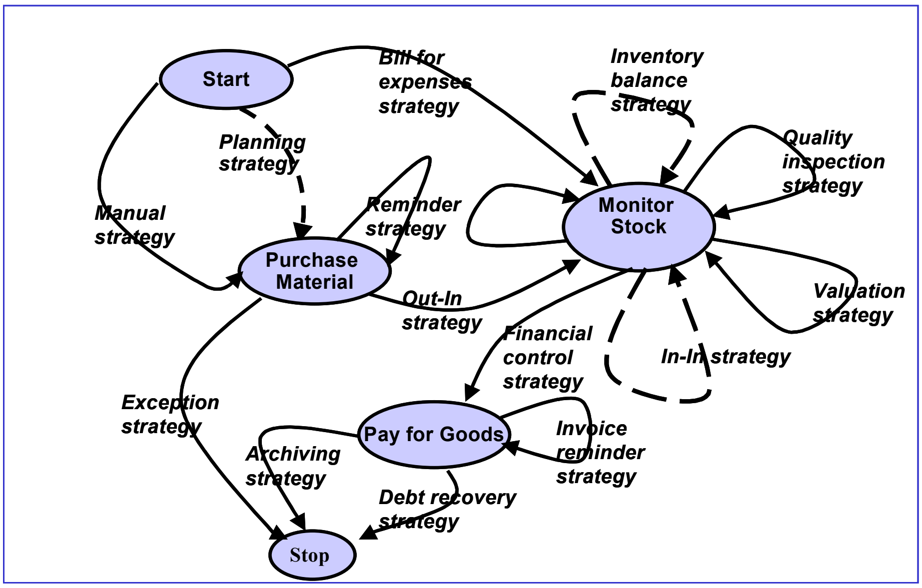

Figure 4. Modified material management map

The new material management map includes an *Exception* strategy, leading from the *Purchase Material* intention to the *Stop* intention to be taken in cases of failure to receive the goods from the supplier. This strategy includes canceling the purchase order, so the process can start again.
As well, a new intention of *Pay for Goods* is added to the map. The *Financial Control* strategy leads to this intention, and it also has a recursive strategy of *Invoice Reminder*, aimed at assuring that the invoice arrives from the supplier and the process is not held waiting for it.
An *Archiving* strategy leads to the *Stop* intention. The *Archiving* strategy includes archiving the process data (purchase order, payment details, etc.) once the process is

completed. We need to add as well a *Debt recovery* strategy to handle the case where the external event does not occur after the reminder strategy has been taken.

5 Conclusion

By combining the map and GPM we arrived at a formalism which precisely defines how human intentions drive a process. The result is an intention driven approach to process modeling that supports the analysis and verification of a designed map.

Modeling a business process in map terms provides an intentional view of what the process aims to achieve and the different ways to do it. Intentions in the map express goals to be attained and hide the details of how to implement them. Strategies are made explicit thus showing the different ways of achieving a goal. Finally, the map, as a multiple assembly of goals with multiple ways of achieving them, represents multiple variations in a business process.

Transforming a business process representation in map terms into GPM concepts provides a state-based formalisation of a map which is conducive to analysis and verification. We showed how reasoning on intention and section sets of states can help identifying incompleteness in the map specification. The basis of this reasoning is the clear distinction between the ultimate goal of an intention and its basic set, where some actions are already done, but the goal is not reached yet. This distinction led to the section classification presented in the paper, which corresponds to the validity analysis guidelines of GPM. Since the reasoning is semantic rather than technical, it is applicable to process models at a variety of scales.

However, it is clear that process analysis needs to be guided and we expect to lay down guidelines to help detecting anomalies and deficiencies in the represented process. It seems also, from the experiments that we conducted, that generic patterns for corrective actions in given deficiency situations could be designed. This will form the topic of future work.

References

[1] ASAP World Consultancy and J. Blain et al, *Using SAP R/3,* Prentice Hall of India, 1999.

[2] Bider, I., Johannesson, P., Perjons, E. (2002), “Goal-Oriented Patterns for Business Processes”, Position paper for Workshop on Goal-Oriented Business Process Modeling (GBPM’02).

[3] Bunge. M., *Treatise on Basic Philosophy: Vol. 3, Ontology I: The Furniture of the World.* Reidel, Boston, 1977.

[4] Bunge. M., *Treatise on Basic Philosophy: Vol. 4, Ontology II: A World of Systems*, Reidel, Boston, 1979.

[5] Dietz, J.L.G., Basic Notions Regarding Business Processes and Supporting Information Systems, Proceedings of BPMDS’04, *CAiSE’04 Workshops Proceedings*, Latvia, Riga, Vol. 2, pp. 160-168, 2004

[6] Hammer, M. and Champy, J. (1994), *Reengineering the Corporation – A manifesto for Business Revolution*, Nicholas Brealey Publishing, London.

[7] Ould, M. A., *Business Processes: Modeling and Analysis for Reengineering and Improvement*, John Wiley & Sons, 1995.

[8] Paulson D. and Wand, Y., 1992, "An Automated Approach to Information Systems Decomposition," *IEEE Transactions on Software Engineering*, 18 (3), pp. 174-189.

[9] C.Rolland, N. Prakash, A. Benjamen*, A Multi-Model View of Process Modelling,* Requirements Engineering Journal, 4(4) pp 169-187, 1999

[10] Rolland C., Prakash N., *Bridging the gap between Organizational needs and ERP functionality*. Requirements Engineering Journal 5, 2000.

[11] C. Salinesi, C. Rolland. *Fitting Business Models to Software Functionality: Exploring the Fitness Relationship*. 15th Conference on Advanced Information Systems Engineering, (CAISE'03), Springer-Verlag (pub), 2003.

[12] Soffer, P., and Wand, Y., 2003, "On the Notion of Soft Goals in Business Process Modeling", *Business Process Management Journal* (to appear).

[13] Soffer P. and Wand Y., 2004, Goal-driven Analysis of Process Model Validity, *Advanced Information Systems Engineering (CAiSE'04)* (LNCS 3084), p. 521-535

[14] Wand, Y. and. Weber, R (1990), "An Ontological Model of an Information System", IEEE Transactions on Software Engineering, Vol. 16, No. 11, pp. 1282-1292.

[15] Yu, E., and Mylopoulos, J. (1996), "Using Goals, Rules, and Methods to Support Reasoning in Business Process Reengineering", International Journal of Intelligent Systems in Accounting, Finance and Management, Vol. 5, pp. 1-13.