Business Process Lines to deal with the Variability

Colette Rolland [1]
[1] Université Paris 1Panthéon Sorbonne
Centre de Recherche en Informatique
90, rue de Tolbiac 75634 Paris cedex 13 France
Colette.Rolland@univ-paris1.fr

Selmin Nurcan [1,2]
[2] IAE de Paris - Sorbonne Graduate Business School
21, rue Broca 75005 Paris France
Selmin.Nurcan@univ-paris1.fr

Abstract

Variability proved to be a central concept in different domains, manufacturing, software development etc. in order to develop solutions that can be easily adapted to different organizational settings and different sets of customers at a low price. We argue that families of business process models can facilitate the installation of situated models in different organizations. We propose a representation system called Map to capture variability in process models expressed in an intentional manner through business goals and strategies. The paper presents an intentional view of the business process variability and illustrates it in an excerpt of a real case in the Electricity Supply Industry. The kernel of an organizational level model is also introduced to allow the organizational implementation of each variant.

1. Introduction

In all management challenges, one of the major requirements related to the information systems is to be continuously adapted to changing business practices and needs. This can be achieved by developing process-centric solutions. The importance of establishing and preserving the 'best fit' between organization needs (why and what) and system functionalities (how), i.e. between *process models* and *IS specifications* is commonly accepted today.

1.1. Background

Most of the models provided in the literature [5], [10], [15], [20], [36] concentrate on *Who* does *What*, *When*, i.e. on the description of the operational performance of tasks to produce results. Despite the fact that process modeling appears to be a corner stone to help managers improve operational performance, it demonstrated to be insufficient to help organizations in a constantly changing environment. Among others, [37] argues that a more systemic view of an organization is necessary to handle the problem 'in the large' and suggests abstracting from the details of process models in a goal model. A similar *goal-driven perspective* as part of an holistic view of organizational knowledge can be found in a number of approaches such as EKD-CMM [3], [32] and to support business process reengineering [1], [17], [27], [45], as well as in approaches to requirements engineering. The concern is to establish a close relationship between the '*Why'* and the '*What'*. The former captures the strategic goals of the organization whereas the latter tells us how they are achieved through tasks carried out by actors. Focusing on the '*Why'* is essential to avoid unnecessary details and to focus on the 'essence' of the business, on what needs to be achieved and the strategies required to achieve it independently on how to organize the business in order to achieve it (the '*What'*, '*Who'*, '*When'*). This relationship is also of prime importance to handle change and propagate intentional changes onto organizational ones.

1.2. Variability in process families

Over the two last decades, market changes have led to a business environment that is constantly evolving. Companies change to better satisfy customer requirements, improve internal processes and adapt their products and services. The process wave initiated by Hammer and Champy [13] led to the creation of large portfolios of business process models. We continue to develop business process models since they are recognized as indispensable artifacts to drive business management and evolution. These portfolios evolve due to internal factors leading to business process evolution and/or external factors or mergers and acquisitions where different processes, perhaps having common parts, have to be integrated. Further, business process change involves the reuse of parts of the process to be discarded, inclusion of parts of other processes, co-existence of different versions of the same process etc.

The situation is similar to that in manufacturing and product engineering where the notions of product lines and product families have been introduced. *By analogy, we recognize business process lines and families in organizations of today.* Design of product lines and families demonstrated the need to elicit commonalities as well as variable parts in a family and stressed the importance of the variability concept. Variability has been introduced to explicitly differentiate between the common and different parts in a set of similar but different product lines of a product family. Managing commonalities and variability leads to two major advantages:

(a) reuse of common parts [26], [44] and,

(b) adaptation of products to different customers and various organizational settings [42]

Seeing the duality that exists between products and processes, we believe that business process families could beneficially be handled by introducing the concept of variability. The foregoing suggests a move away from management of individual process to managing a set of similar processes considered as a whole, a family. We propose to organize business processes as business process families and to manage variability and commonalities within the family in order to promote reuse and adaptability of business process models.

We understand a business process family to be a collection of processes meeting a common goal but in different ways. For example, the goal 'admit students' can be achieved through a business process family comprising three processes that select students on the basis of a national entrance examination, a university test, or school performance respectively. The variability across the three processes is obvious. However, there is a commonality between these three processes as well: all these processes have to accept fees from the admitted student.

In this paper, we propose a modeling formalism called MAP to capture variability across business processes of a family in an intentional manner. The map is a directed, labeled, non-deterministic graph with goals as nodes, and strategies to achieve goals, as edges. Its nature allows the capture of different forms of variability through multi-edges between a pair of nodes thereby enabling many different traversals of the graph from beginning to end. Besides, using the refinement mechanism of the map, it is possible to represent variability at different levels of detail, in a hierarchy of maps. We show that this hierarchical nature permits us to represent process families as maps. We also show the power of a map to represent variability and, as an illustration, model the variations of an electricity supply process family as a hierarchy of maps.

The paper is organized in five sections. The next section summarizes the flexibility requirements for business process modeling. Section 3 introduces the MAP formalism to capture variability of business processes modeled in an intentional manner. Section 4 presents an example and also considers the adaptation of a business process model within a family. Section 5 introduces some kernel concepts for the organizational implementation of the business variants.

2. Business process modeling and flexibility requirements

Flexibility has been the focus of many researches [30], [35], [38], [41] and many definitions can be found in literature. Processes flexibility means fast reactivity to internal and external changes. It also reflects the easiness to make evolve business process models.

Literature provides various process modeling formalisms, which can be classified into four categories: *activity oriented*, *product oriented*, *decision oriented and conversation oriented models*. Each category has its underlying paradigm more or less appropriate for flexible process modeling. We presented a comprehensive survey of this literature in [25]. We consider that a given modeling approach or language can be classified in one or more among those categories.

Activity-oriented models allow us to prescribe a process as a set of activities and their relationships regarding pre-defined control and data flows [14], [20]. Such a process 'programming' allows one to represent the well-defined parts of processes and not the creative parts [18]. *Product-oriented models* put forward the result of the activities of a process. They model the evolution of the product and couple the product state to the activities that generate it [14]. These models seem to be more appropriate than activity-oriented ones for representing knowledge intensive processes.

The most recent type of process models [8], [11], [23], [24] is based on the *decision-(or intention) oriented paradigm* according to which the successive transformations of the product are looked upon as consequences of decisions. Those models are semantically more powerful because they explain not only how the process proceeds but also *why* [21]. This paradigm seems to be particularly appropriate for representing knowledge intensive business processes or any kind of processes requiring flexibility [24].

Conversation models are based on the speech act theory and on the principle that each sentence expressed by someone represents an intention, a commitment [40]. As a matter of fact, highlighting the organizational reasons, which cause the communications between some roles, can justify changes of processes.

Since the introduction of the *process centered view* of organization management by M. Hammer and J. Champy, business process modeling gained importance in both the management community and the system engineering community. During the two late decades, several languages dealing with business process modeling were proposed: traditional input-process-output languages, conversation-based languages, languages based on role modeling, system thinking and system dynamics techniques, and constraint-based languages.

Recent business process modeling languages provide mainly concepts for activity-oriented and product-oriented representations. The Unified Modeling Language (UML) allows modeling application structures, behaviors and architectures, but also processes. Event-Driven Process Chains (EPC) represent temporal and logical dependencies between functions, events or connectors which are linked via control flows. The Business Process Modeling Notation (BPMN) defines a BP model as a network of activities and control flows. The Business Process Management Initiative (BPMI) developed open specifications to enable the standards-based management of cross-enterprise processes based on BPM Systems.

We argue that a business process model must capture much more than the steps of 'procedures'. The concept of goal expresses an intention. The more recent *goal-oriented* approaches [22], [23] define stable characteristics of the business that any organization choice must respect but also the variation points. Compound goals can be decomposed into sub-goals. At the most detailed level, the way of achieving the atomic goals may be specified in terms of actors, activities and control flows, i.e. organizational and operational concerns. We argue that knowledge intensive business processes of modern organizations require modeling artifacts that can represent the organizational goals, strategies, responsibilities and risks rather than exclusively actors, operations and activities.

3. Capturing business variability in Maps

We use the MAP formalism [33] to capture variability of business processes modeled in an intentional manner.

3.1. Variability and goals

We understand a business process family to be a collection of processes meeting a *common goal* but in *different ways*. We consider a family of processes to deal with material management. The family comprises a large number of process elements such as Material Purchasing, Stock Inventory, Receiving stock, Inspecting stock etc. all aiming at contributing to the achievement of the family goal '*Satisfy Material Need Efficiently'*. Every element of the family looks for the achievement of its own goal as a contributor to the main common goal. These are respectively, *Purchase Material, Conduct Inventory, Receive Stock,* and *Inspect Stock* for the aforementioned process elements.

What we find specific to goals related to a business process family is that they are associated with *different ways of achievement* that we refer to as *strategies of achievement*. In this sense, we can say that family goals are multi-facetted in contrast to mono-facetted goals in a single business process modeling perspective. For example, for the goal *Purchase Material*, in one case it might be considered enough to know that an organization achieves this goal by *Forecasting material need.* Purchase material is mono-facetted: it has exactly one strategy for its achievement. However, in the new family context, it is necessary to introduce other strategies as well, say the *Reorder Point* strategy for purchasing material. *Purchase Material* now is multi-facetted; it has many strategies for goal achievement. The variability across the associated business process elements to the goal *Purchase Material* is obvious. However, there is a commonality between these three processes as well: all these processes have to fill in the purchase order, edit it, have it signed and sent.

The multi-faceting of goals appears at every level of family elements and at the level of the family itself as well. In the material management family, there are many ways to achieve the common goal '*Satisfy Material Need Efficiently' :* (a) by direct entry in stock followed by stock monitoring and its alternative (b) the path composed of Purchasing material, Receiving stock of purchase material and Monitor stock.

3.2. Levels in variability modeling

Variability in business process families arises from the fact that different members of the family of business processes can achieve goals of the family in different ways. This variability is in the context of an intentional view of a process. However, variability can be looked upon also at relatively more organizational and operational level of flow of activities or state transitions, for example (see § 5). This leads to modeling variability at two levels, the organizational and intentional (purpose driven) levels as shown in Figure 1.

This finds support in the literature where different types of process formalisms can be classified depending on the role that is played by goals. 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" [13]). 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 [9]. Activity-sequence oriented languages (e.g., UML Activity Diagram), agent-oriented languages (e.g., Role-Activity Diagram [27], state-based languages (e.g. UML state charts) belong to this category. 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 such as Map [33], follows the human intention of achieving a goal as a force which drives the process [41]. As a consequence, goals to be accomplished are explicitly represented in the process model together with the alternative ways for achieving them, thus facilitating the selection of the appropriate alternative for achieving the goal.

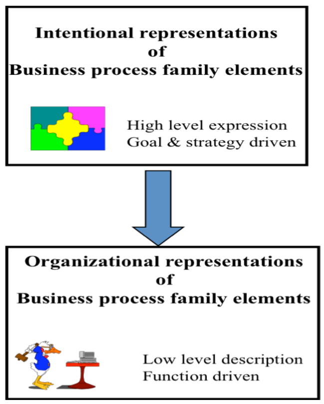

Figure 1. Levels of Variability

We develop our view by considering the *intentional level of variability modeling* and therefore select a formalism at this level. We propose to use the modeling formalism called MAP to capture variability in an intentional manner. In the following, we show that it is indeed possible for the map to represent the variability at the intentional level.

A survey of flexible/adaptive workflow approaches situates the properties of the studied approaches [25] with respect to the workflow applications life cycle (build time and run time). Among these properties, the *nature of the flexibility* defines if the capacity of taking into account the environmental change may be incorporated in the process model during the build-time or not. The flexibility by adaptation (a posteriori) allows adapting the process definition or its instances during their execution. This is the most usual case in the literature. Approaches that offer only this kind of flexibility are based on prescriptive modeling formalisms and are mainly activity-oriented. These approaches cannot anticipate the capacity to change during the build-time. This concerns most of the workflow modeling languages. On the opposite, the flexibility by selection (a priori) is based on modeling formalisms that can offer the capacity to deal with the environmental change without any evolution of process definitions. This means that this capacity may be incorporated in process definitions during build-time. The Map model, as well as few other approaches in the literature offer this capability, for instance [6].

We will see in the next sections that the Map model is close to declarative modeling that avoids rigid workflow challenges providing the users with more flexibility. It also allows a deferred design approach; i.e. it allows deferring the specific navigation choices to run time (see run-time adaptation in §4.2).

3.3. Business intentionality in Maps

A map is a process model expressed in a goal driven perspective. Map provides a process representation system based on a non-deterministic ordering of goals and strategies. A map is represented as a labeled directed graph (see example in Figure 7) with goals as nodes and strategies as edges between goals. The directed nature of the graph shows the achievement of which goals can follow the achievement of which others.

A *Goal* can be achieved by the performance of a process. Each map has two special goals, *Start* and *Stop* to start and end the process respectively. A *Strategy* is an approach, a manner to achieve a goal. A strategy S_{ij} between the couple of goals G_i and G_j represents the way G_j can be achieved once G_i has been satisfied.

A *Section* is a triplet <G_i, G_j, S_{ij}> and represents a way to achieve the target goal G_j from the source goal G_i following the strategy S_{ij}. Each section of the map captures the situation needed to achieve a goal and a specific manner in which the process associated with the target goal can be performed. A section in a map can be *refined* as a map (see Figure 8). This leads to intentional process modeling as a hierarchy of maps. The refinement continues until the operational level is reached. At this level, the body of the section describes an operational process (the "internal" view of a process, i.e. the flow of activities) leading to the construction of some product.

3.4. Modeling business variability in Maps

For the sake of conciseness, we use a textual notation in which goals are named by letters of the alphabet, strategies are numbers and therefore, a section named ab_i designates a way to achieve a target goal *b* from a source one *a* following a strategy *i*. Thus, the section <G_i, G_j, S_{ij}> is named ab_1 where *a* is the code of the goal G_i , *b* is the code of the goal G_j and *1* is the code of the strategy S_{ij} (see Figure 2).

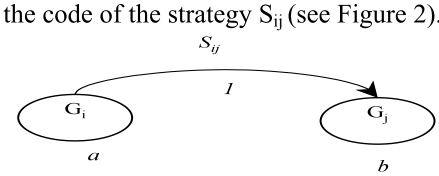

Figure 2. A section

We advocate that sections are at the right abstraction level to capture business variability. We consider a section as an important process characteristic that business agents (managers, decision makers, actors…) want the business to provide and also an abstraction of a business flow. By analogy with software variability, a section can be related to the notion of a feature. In FODA [16] for example, a feature is defined as *"a prominent or distinctive user-visible aspect, quality or characteristic of a software system or systems"*. In [4], a feature is *"a logical unit of behavior that is specified by a set of functional and quality requirements"*. The point of view taken in this paper is that a *business feature* is a representation of a visible process characteristic and an abstraction of a cohesive business flow of activities expressed in an intentional manner.

Features represented in a map are related to each others by four kinds of relationships namely *multi-thread*, *bundle*, *path* and *multi-path* relationships. The relationships show the possible combination of features from which a business agent can select the appropriate ones according to the situation at hand. Let us now see how these relationships are used to express variability in business models.

The multi-thread relationship: when there are various ways to achieve the same goal starting from a source, features are related by a multi-thread relationship. A multi-thread relationship is represented in a map by several strategies between a pair of goals as represented in Figure 3. It shows through the strategies the different flows of activities provided to obtain the same result.

A multi-thread relationship expresses a business feature variability by grouping optional features from which one or many features can be selected.

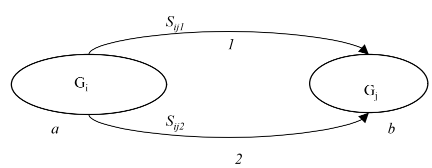

Figure 3. A multi-thread relationship

The Bundle relationship: In the case where the several ways to satisfy the same goal are exclusive, we relate them with a bundle relationship. It implies that only one way can be selected to achieve the target goal. Figure 4 shows an example of a bundle relationship. The bundle relationship expresses a feature variability by grouping alternative features that are mutually exclusive.

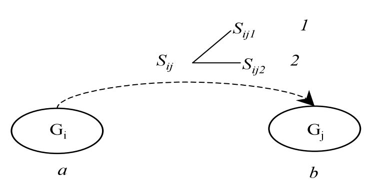

Figure 4. A bundle relationship

The Path relationship: when the achievement of a target goal *b* from a source goal *a* requires the satisfaction of intermediary goals, we introduce a path relationship. It establishes a precedence/succession relationship between features expressing that in order to trigger a business flow, some other business flow must be executed first. In general, a path relationship is a composition of features, features related by multi-thread or bundle relationships or other paths. Some paths can be iterative.

Figure 5 represents a path relationship between the goals G_i and G_k, denoted respectively by a and c, which is composed of the multi-thread relationship containing the features ab1, ab2, ab3, and the feature denoted bc1. It expresses that in order to achieve the goal G_k , we must first select and execute one or many features among ab1, ab2 or ab3 and then execute the feature bc1.

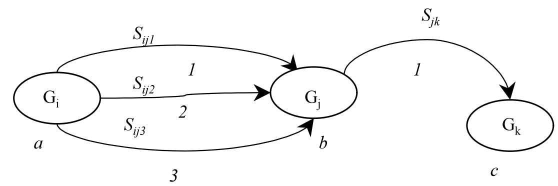

Figure 5. A path relationship

Multi-path relationship: given the multi-thread, bundle and path relationships, a goal can be achieved by several combinations of strategies. This is represented in the map by a pair of goals connected by several sections. Such a relationship is called a multi-path relationship. For example, we show in Figure 6 two alternative paths to satisfying the goal G_k (denoted c) starting from the goal G_i (denoted a). The first path achieves G_k through the intermediary goal G_j whereas the second path achieves G_k directly from G_i.

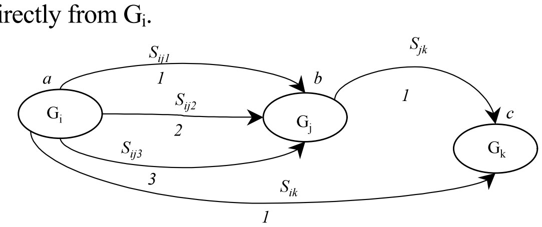

Figure 6. A multi-path relationship

A multi-path relationship identifies the several combinations of business flows (represented by paths of sections) that can be executed to satisfy the same goal.

Thus, *a multi-path relationship is a means to express business feature variability by grouping the alternative paths satisfying the same goal.*

In general, a map from its *Start* to its *Stop* goals represents all possible combinations of features expressed by multi-thread, multi-path and bundle relationships. Each particular combination of features is a path, from the *Start* goal to the *Stop* one, that describes a way to reach the final goal *Stop*.

3.5. Generating variants embedded in a Map

We notice that the bundle and multi-thread relationships are easily visible in the map. However, it is more difficult to identify all the combinations of features in a map (based on multi-path and path relationships). We propose to apply McNaughton and Yamada's algorithm [19] in order to discover systematically all the paths embedded in a map. The algorithm is based on the two following formula:

Let s and t be the source and target goals, Q the set of intermediary goals including s and t and P the set of intermediate goals excluding s and t.

The initial formula $Y_{s,Q,t}$ used to discover the set of all possible paths using the three operators that are the union ("$\cup$"), the composition operator (".") and the iteration operator ("") is :*

$$Y_{s,Q,t}=(X_{s,Q\backslash\{s\},s})^{*}.\ X_{s,Q\backslash\{s,t\},t}\ .\ X^{*}_{t,Q\backslash\{s,t\},t}$$

And given a particular goal q of P, the formula $X_{s,P,t}$ applied to discover the set of possible paths is :

$$X_{s,P,t}=X_{s,P\backslash\{q\},t}\cup X_{s,P\backslash\{q\},q}.\ (X_{q,P\backslash\{q\},q})^{*}.\ X_{q,P\backslash\{q\},t}$$

W specialize the $X_{s,P,t}$ into paths, multi-paths, multi-threads and bundle relationships that we note as follows:

Multi-thread relationship between two goals k and l is denoted: $MT_{kl}=\{kl_1 \vee kl_2 \vee ... kl_n\}$ where the kl_i are the features related by the multi-thread relationship. Thus, the multi-thread represented in Figure 3 is : $MT_{ab}=\{ab_1 \vee ab_2\}$

Bundle relationship between two goals k and l is denoted: $B_{kl}=\{kl_1 \otimes kl_2 \otimes ... kl_n\}$ where the kl_i are the exclusive features related by the bundle relationship. In Figure 4, the bundle relationship is : $B_{ab}=\{ab_1 \otimes ab_2\}$

Path relationship between two goals k and l is denoted $P_{k,Q,l}$ where Q designates the set of intermediary goals used to achieve the target goal l from the source goal k. A path relationship is based on the sequential composition operator "**.**" between features and relationships of any kind. As an example, the path relationship of Figure 5 is denoted: $P_{a,\{b\},c} = MT_{ab}.\ bc1$

Multi-path relationship between two goals k and l is denoted $MP_{k,Q,l}$ where Q designates the set of intermediary goals used to achieve the target goal l from the source one k. A Multi-path relationship is based on the union operator "$\cup$" between alternative paths. Thus, the multi-path of Figure 6 is denoted : $MP_{a,\{b\},c} = P_{ac} \cup MT_{ab}.P_{bc}$

In section 4 we illustrate the variability approach with an excerpt of a real case example and show the application of the MacNaughton-Yamada's algorithm. However, due to paper length limit, we present only the results obtained after applying the algorithm. It can be seen that the goal of the business process family is captured in a *hierarchy of maps*. The goal associated to the root map is the high level statement about the purpose of the family. Using the refinement mechanism each section of the root map can be refined as a map and the recursive application of this mechanism results in a map hierarchy. At successive levels of the hierarchy, the goal stated initially as the goal of the root map is further refined. At any given level of the hierarchy, a map describes the business process family as a set of business features and feature variability through four types of feature relationships. Multi-thread and bundle introduce local variability in the sense that they allow to represent the different ways for achieving a goal directly. Path and multi-path introduce global variability by representing different combinations of business features to achieve a given map goal. Any path from Start to Stop represents one way of achieving the map goal, therefore the purpose represented in this map.

4. An illustration

To illustrate our approach, we consider a business example from the ESI (Electricity Supply Industry) sector. The example is simplified to meet the paper size requirement but extracted from a real three-years project involving three large European Electricity companies and our research group [12]. The map of Figure 7 provides the top-level intentional view of the electricity supply business family to support Electricity Supply Management (ESM).

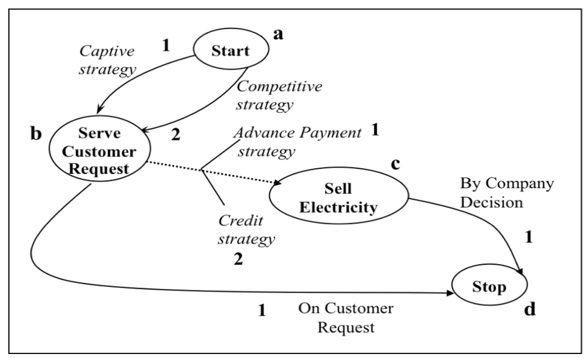

Figure 7: Map sample

4.1. Business Process Family Modeling

Figure 8 shows a refined view of the section bc2 of this map itself expressed as a map. The map shown in Figure 7 is organized around two key goals, *"Serve Customer Request"* and *"Sell Electricity"* that represent generic goals in the sense that they exist in any electricity distribution process. Furthermore, the map indicates an ordering constraint: in order to sell electricity to a customer, his/her request for electricity provision has to be fulfilled first.

In the ESM map, it shall be noticed that there are two different strategies to achieve each of these two goals. For example, the *"Advance Payment strategy"*, and the *"Credit strategy"*, are two alternative strategies to achieve the business goal " *Sell Electricity"*. These map strategies identify two rather different business strategies to get the customer to pay for his electricity consumption. Indeed the *"Advance Payment strategy"* refers to a solution based on the use of payment cards to energize the customer meter whereas the *"Credit strategy"* refers to the more conventional solution where the electricity company provides electricity to its customer and gets paid after consumption.

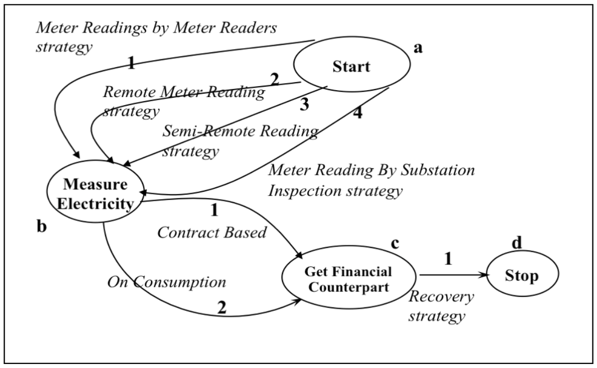

Figure 8: Map refining bc2 section

Each section in the map represents a *feature* that the business process family can provide. Further, this example demonstrates the *feature variability* in the ESM family that is captured by the map. We distinguish two kinds of variability that are :

(i) a variability in strategies provided to satisfy the same goal and,

(ii) a variability in the combinations of strategies to fulfil the same goal.

The first kind (i) is expressed by the multi-thread or the bundle relationship. In our example, we depict a bundle relationship between the couple of goals "*Serve Customer Request*" and "*Sell Electricity*" respectively denoted *b* and *c* and composed of the two exclusive features *bc1* and *bc2* corresponding to the sections <*Serve Customer Request, Sell Electricity, Advance payment strategy*> and < *Serve Customer Request, Sell Electricity, Credit strategy* >. We also identify a multi-thread relationship composed of the features ab1 and ab2 corresponding to the sections <*Start, Serve Customer Request, Captive strategy*> and < *Start, Serve Customer Request, Competitive strategy* >.

The second kind (ii) is expressed by the multi-path relationship. It shows the different combinations of business flows that can be executed to satisfy the same goal. For example, given an electricity connection obtained after achieving the goal "*Serve Customer Request*", we can follow two alternative paths to stop the process. We can either respond to the customer demand applying the "*On customer request strategy*" or we can proceed with consumed electricity billing and payment through either the *"Advance payment strategy"* or the *"Credit strategy"* then stop *"By company decision"* if payment is not made after a given delay.

In order to identify all the combinations of features, we apply the MacNaughton-Yamada's algorithm introduced in Section 3.5. The initial formula generating all the paths between the goals a and d is : $Y_{a\{a, b, c,, d\},d}=(X_{a, \{b,c,d\}, a})^*. X_{a,\{b,c\},d} . X^*_{d,\{b,c\},dt}$ The identified paths (and therefore composition of features) are summarized in Table 1.

Table 1. List business features and their composition

Business features	ab_1 ab_2 bc_1 bc_2 cd_1 bd_1
Feature composition kind	Identified compositions
Path	$P_{a,\{b,c\},d} = P_{ab} . MP_{b\{c\}d}$ where $P_{ab} = MT_{ab}$
Multi-Path	$MP_{b\{c\}d}=P_{bd} \cup B^*_{bc} . P_{cd}$ $P_{cd} =cd_1$ $P_{bd}=bd_1$
Bundle	$B_{bc}=bc_1 \otimes bc_2$
Multi-thread	$MT_{ab}=ab_1 \vee ab_2$

In the next section we will discuss how the map representation can help in customizing a business process to specific needs.

4.2. Adapting a Business Process

Since a map captures a full range of features permitted in a family, the adaptation issue is of determining which features and which combination of features are relevant to the business process under design. There are two kinds to adaptation:

- *Design time adaptation* permits a selection of a combination of features that results in only one path from Start to Stop.
- *Run time adaptation* allows to leave a large degree of variability in the adapted map and the desired features can then be selected dynamically at enactment time of the process.

It is possible for business people to perform this adaptation. This is because a knowledge of the business characteristics and an analysis based on these is enough to make the adaptation decision. To illustrate this aspect, we perform pay-off analysis on the map of Figure 8. The features that form part of the adapted map are determined by an analysis of the benefits that accrue from features standing alone and in combination with other related features. To adapt section bc2 of the map of Figure 7, one has to decide on how electricity should be measured and how the financial counterpart should be obtained. This leads to selecting the appropriate features and feature combinations of the map presented in Figure 8. Each feature selection has however a payoff that can be analyzed in the view of its combination to another one. The pay-off analysis for bc2 features is summarized in the Table 2 below.

Let us consider the case where it is necessary to get financial counterparts both contract based and on consumption. Table 2 shows that remote readings are a cost effective way to handle measurement in both cases. Indeed, it is real time and adapted to payment on consumption. Besides, the cost of installing remote readers can be included in the contract prices and recovered in the long term. However, the payoff table also says that remote reading, as it is automated, is not fully reliable and should be double-checked, e.g. by using substation inspection. One possible adaptation of section bc2 is then to keep the features ab2 and ab4 along with bc1 and bc2.

Table 2. Pay-off summary

		Get financial counterpart	
		Contract based	**On consumption**
Measure electricity consumption	**Meter reading by meter reader**	Can be envisaged at sustainable cost if visits are achieved at a low frequency e.g. once or twice a year)	Excluded because too difficult to organize all visits at the required pace.
	Remote reading	Cost effective combination that can be done in real time. However, remote reading is not completely secure. A complementary check of electricity measurement is thus needed, e.g. by meter reader, or by substation inspection.	
	Semi-remote reading	Cost effectiveness is a linear function of the number of contracts per cluster of semi-remote reader.	Very costly if the number of customers paying on consumption, per cluster of remote reader is low.
	Substation inspection	Only possible if the connected meter readers relate to single contract. Otherwise, calls for individual reading.	Cost effective way to handle the verification of consumers invoiced by remote reading clustered on the same substation.

5. Organizational implementation of the business features

Service-oriented Computing is the computing paradigm that utilizes services as fundamental elements for developing software applications [28]. We suggest using a similar paradigm to specify and than implement the business features and their variants previously identified at the intentional level.

The Service-oriented Enterprise implies that business centric organizations offering business services shall describe their services in an intentional manner as developed above, and publish them to a business service registry that makes these descriptions available. Business agents, with a decisional role, (management, control, compliance) who are looking for services, use an intention matching mechanism to retrieve service descriptions.

Business service description can also include variability at an organizational level (roles, preferences of actor, context awareness), i.e. propose alternative organizational variations of a business service. The ability to integrate the context related knowledge at the organizational level, allows business process models to be active, flexible, fine-grained and able to express a variety of business rules. In [38], a role driven approach has been developed for supporting context awareness with two major benefits: (i) flexibility in assigning functions to roles since a function can be performed by several possible roles according the performance context rather than a specific one, and (ii) autonomy given to actors allowing them to develop strategies for performing operations, operational goals and functions.

In the domain of the enterprise modeling, it is a common way to consider that operationalizable business intentions are implemented using business process models [1], [3], [45]. We consider that a *business process chunk* [23] operationalizes a map section, which cannot be refined any more using intentional/decisional considerations (see Figure 9). In other words, a business process chunk defines the internal structure of a business service by the means of a flow of activities leading to the construction of the desired product and/or service.

At this layer, *roles* that describe the ability to act in order to achieve *business intentions* according to *strategies* associated to *sections*, the *actors* holding these *roles*, the *activities* they will perform and the pre-order of these activities (when the business process is well-defined) should be defined (Figure 9). *Actors* perform *activities* that specify the smaller work steps in a business process. A comprehensive version of this organizational model is presented in [23].

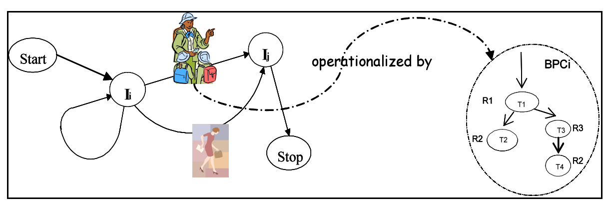

Figure 9. Articulation between intentional and organizational layers

The representation formalisms used at this level can be classical activity-oriented models. The purpose of the Map Model, which is proposed to analyze business at the intentional layer, is thus to define the *integration/ orchestration* for all those *islands of business process chunks and to specify all possible variants for the business features*.

6. Conclusion

We have proposed here a move from individual business process modeling to the modeling of entire families of business process models. The process models that comprise this family come together with their own goals but the combined effect is that of achieving the goal of the family. From the perspective of the family, the important dimension to modeling is the bringing together of the various variants in one coherent whole. These variants arise because of the diversity in the ways in which goals can be achieved, the different strategies available for goal achievement.

For modeling families in their full variability, we found the power of the map formalism to be just right. It explicitly recognizes goals and strategies as key concepts and provides a range of inter-variant relationships. Using these, we expect to base the representation of business process model families in the future.

Families have a common part and a variant part and in identifying the former, families can be built through reuse, a property that we shall explore in the future. Additionally, since families have the potential to model variants, they have the capacity to handle diverse models. This capacity can contribute to handling different customers and organization settings.

The notion of variability in business process families introduced here brings together a set of similar but different processes to facilitate reuse and adaptation. We use a goal driven formalism that is the Map, to represent business process families as a set of business features and feature variability through four types of feature relationships. Once the process family has been expressed with maps, the task of building the adapted business process model to a given setting can be simply done by deciding which combinations of features are the most suited to the situation at hand. We think that expressing the variability with the map formalism is particularly useful at the adaptation phase. It exposes the business process owner to the choices that are relevant to the satisfaction of his/her goals in terms of the properties of the business and there is no need to deal with organizational and technical configuration details.

The proposed business process modeling approach gives us the ability to describe, initially, the invariants and the variants of the organization in terms of objectives and strategies before specifying the manner of making them operational in a particular organizational situation. The purpose of the Map Model is thus to define the *orchestration* of the *islands of business process chunks*. This coupling is achieved in the map formalism by simply relating each operationalizable section (which does not need further refinement from an intentional perspective) of a map to an activity-oriented representation of the underlying business process chunk (internal view).

As a matter of fact there are other approaches (for instance EPC) to explicitly define variability in business processes. Nevertheless, in our work we do not want to go directly to the details of the business processes but to deal first with their essence (purpose). We argue that modeling business processes without having the right level of abstraction disables our capability to explain the required business changes. We believe in goal modeling as an abstraction tool. One can use the goal modeling to begin the business analysis and then move to more detailed descriptions using appropriate languages (see figure 1 and §5).

The goal modeling can be considered as a challenging issue. In fact, experience shows that (i) goal discovery is not an easy task [2], [31], (ii) application of goal reduction methods [7] to discover component goals of a goal is not as straight-forward as the literature suggests [2], [5], and (iii) eliminating uninteresting and spurious goals is necessary and nevertheless difficult [29]. The map approach offers assistance for goal driven modeling thanks to the mechanisms of intention and strategy formulation and refinement concept [24], [34]. Goal templates provide the potential for introducing guidance in goal formulation.

The Map approach was used over a dozen industrial and research European projects. In DIAC, it helped understanding the move towards a customer centric business. DIAC is the financial branch of the Renault motor, which grants credit to Renault customers and sells other related financial services [34]. The Map approach was used to handle the standardization of practices in the different DIAC subsidiaries located in different countries. Maps were also useful to identify new strategies such as cross-selling, new sales channel, keeping customer loyalty etc. In BNP Paribas, the Map approach has been used to diagnose business/IT strategic alignment, to discover evolution requirements, and finally to propagate and validate these requirements [43]. In Société Générale, the approach guided a systematic research of exceptional business process variants [39]. In [23], the purpose of its usage was to derive an intention driven model for flexible workflow applications.

Future work consists of implementing a configuration tool to adapt a business process model of a family using the map formalism and developing a software tool to support navigation in a map to select dynamically the feature most appropriate to the situation at hand.

7. References

[1] Antón, A. I., McCracken, W. M. and Potts, C. , "Goal Decomposition and Scenario Analysis in Business Process Reengineering", CAiSE '94, The Netherlands, 1994, pp. 94-104.
[2] Antón, A.I., "Goal based requirements analysis", ICRE'96, Colorado Springs, Colorado, 1996, pp. 136-144
[3] Barrios, J., and Nurcan, S. , "Model Driven Architectures for Enterprise Information Systems", CAISE'04, Latvia, Springer Verlag, 2004.
[4] Bosch J., Florijn G., Greefhorst D., Kuusela J., Obbink H., andPohl K., "Variability issues in Software Product Lines", 4th Int. Workshop on Product Family Engineering, Bilbao, Spain, 2001.

[5] Bubenko, J., “Enterprise Modeling”, Ingénierie des Systèmes d'Information, 2(6), 1994.
[6] Dadam, P., Reichert, M. and Rinderle, S., “Evaluation of Correctness Criteria For Dynamic Workflow Changes”, BPM'03, Eindhoven, The Netherlands, 2003, pp. 41-57.
[7] Dardenne, A., van Lamsweerde, A. and Fickas, S., “Goal directed requirements acquisition”, Science of Computer Programming, 20(1-2), 1993, pp. 3-50.
[8] Dellen, B., Maurer, F., and Pews, G., “Knowledge Based Techniques to Increase the Flexibility of Workflow Management”, Data and Knowledge Engineering, 23(3), North Holland, 1997.
[9] Dietz J.L.G., “Basic Notions Regarding Business Processes and Supporting Information Systems”, BPMDS’04 Workshop (CAISE), Latvia, Riga, 2004, pp.160-168.
[10] Eriksson, H.-E., and Penker, M., Business modeling with UML-Business patterns at work, J. Wiley, New-York, 2000.
[11] Faustmann, G., “Enforcement vs. Freedom of Action- An Integrated Approach to Flexible Workflow Enactment”. Workshop on Adaptive Workflow Systems. CSCW, Seattle, USA, 1998
[12] Grosz G., Loucopoulos P., Rolland C., and Nurcan S. “A Frame-work for Generic Patterns Dedicated to the Management of Change in the Electricity Supply Industry”, DEXA '98 Workshops, 1998.
[13] Hammer, M., and Champy, J., Reengineering the Corporation-A manifesto for Business Revolution, Harper business, 1994.
[14] Harel, D., “STATEMATE: A working environment for the development of complex reactive systems”, IEEE Transactions on Software Engineering, 16(4), 1990, pp.403-414.
[15] Jarzabek, S., and Ling, T.W., “Model-based support for business reengineering”, Information and Software Technology, 38(5), 1996, pp. 355-374.
[16] Kang K., Cohen S., Hess J., Novak W., and Peterson A.,“Feature-Oriented Domain analysis (FODA) Feasibility Study”., Technical report CMU/SEI-90-TR-21, Carnegie Mellon University, 1990.
[17] Lee, J. , “Extending the Potts and Bruns Model for Recording Design Rationale”, ICSE’91, Austin, Texas, 1991, pp. 114-125.
[18] Lehman, M.M., Process Models, Process Programs, Programming Support, ICSE’87, 1987.
[19] McNaugthon R., and Yamada H., “Regular expressions and state graphs for automata”, IEEE Transactions on Electronic Computers, EC-9, 1960, pp. 39-47.
[20] Marca, D.A., and McGowan, C.L., IDEF0/SADT: Business Process and Enterprise Modeling, Eclectic Solutions, Inc, San Diego, 1993.
[21] McLean, A., Moran, T., and Young, R.M., “Design Rationale: the argument behind the artifact”. CHI'93, 1989, 247-256.
[22] Neiger, D., and Churilov, L., “Goal-Oriented Business Process Modeling with EPCs and Value-Focused Thinking”, LNCS Vol. 3080, 2004, pp. 98-115.
[23] Nurcan, S. and Edme, M.-H., “Intention Driven Modelling for Flexible Workflow Applications”. Spec. issue of the Software Process: Improvement and Practice Journal on "Business Process Management, Development and Support", 2005, 10(4).
[24] Nurcan, S., Etien, A., Kaabi, A., Zoukar, I., and Rolland, C., “A Strategy Driven Business Process Modelling Approach”, Spec. issue of the Business Process Management Journal on "Goal-oriented business process modeling", Emerald, 2005, 11(6).
[25] S. Nurcan., “A survey on the flexibility requirements related to business processes and modeling artifacts”, HICSS’08, Big Island, Hawaii, USA, 2008.
[26] Ommering R.V., “Building product populations with software components”, ICSE’02, Orlando, Florida, 2002.
[27] Ould, M.A., Business Processes: Modeling and Analysis for Reengineering and Improvement, J. Wiley & Sons, 1995.
[28] Papazoglou M-P., and Georgakopoulos D., “Service-Oriented Computing”, Communications of the ACM Journal, 46(10), October 2003, pp. 24-28.
[29] Potts, C., “Fitness for Use: The System Quality that Matters Most”, REFSQ'97, Barcelona, Spain, 1997, pp. 15-28.
[30] Regev, G., and Wegmann, A., “A Regulation-Based View on Business Process and Supporting System Flexibility” BPMDS’05 Workshop (CAISE), 2005, pp. 91-98.
[31] Rolland, C., Souveyet, C. and Ben Achour, C., "Guiding Goal Modelling using Scenarios", IEEE Transactions on Software Engineering, Special Issue on Scenario Management, 1998, 24(12), pp. 1055-1071.
[32] Rolland, C., Loucopoulos, P., Kavakli, V., and Nurcan, S., “Intention based modelling of organisational change: an experience report”, EMMSAD'99 (CAISE), Germany, 1999
[33] Rolland, C., and Prakash N., “Bridging the gap between Organizational needs and ERP functionality”, Requirement Engineering Journal, Vol.5, 2000, pp180-193.
[34] Rolland, C., Salinesi, C., and Etien, A., "Eliciting Gaps in Requirements Change", Requirements Engineering Journal, 2004, 9(1), pp. 1-15.
[35] Rosemann, M., and Recker, J., “Context-aware Process Design Exploring the Extrinsic Drivers for Process Flexibility”, BPMDS’06 (CAISE), Luxembourg, 2006.
[36] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and W. Lorensen , Object Oriented Modeling and Design, Prentice-Hall, New-York, 1991.
[37] Rummler, G.A. and Brache, A. P., Improving Performance, Jossey-Bass Publishers, Indianapolis, 1995.
[38] Saidani, O., and Nurcan, S., “A Role-Based Approach for Modelling Flexible Business Processes”, BPMDS'06, (CAISE), Luxembourg, 2006.
[39] Salinesi, C. and Ivankina, E. , "Guiding Use Case Discovery By Analysing Threats", CAISE’04, Springer Verlag, Riga, Latvia, 2004.
[40] Medina-Mora, R., Winograd, T., Flores, R., and Flores, F., “The Action Workflow Approach to Workflow Management Technology”, CSCW’92, Toronto, Canada, 1992.
[41] Soffer, P., and Rolland, C., “Combining Intention-Oriented and State-Based Process Modelling“, ER’05, 2005, pp. 47-62.
[42] Svahnberg, M., Van Gurp J., and Bosch J., “On the notion of variability in Software Product Lines”, Working IEEE/IFIP Conference on Software architecture, 2001.
[43] Thevenet, L.-H., and Camille S.,. “Aligning IS to organization’s strategy: the INSTAL method”, CAISE’07, Trondheim, Norway, 2007.
[44] Thomphson J., and Heimdalh M.P., “Extending the Product Family Approach to support n-Dimensional and Hierarchical Product Lines”, 5th IEEE Symposium on Requirements Engineering, Toronto, Canada, 2001.
[45] Yu, E.S.K., and Mylopoulos, J., “Using Goals, Rules and Methods to Support Reasoning in Business Process Reengineering”, Intelligent Systems in Accounting, Finance and Management, Vol. 5, 1996