Variability in Business Process Families

Colette Rolland, Université Paris 1 – Panthéon – Sorbonne, Centre de Recherche en Informatique, 90 rue de Tolbiac, 75634 Paris cedex 13, France; E-mail: rolland@univ-paris 1.fr

Naveen Prakash, Université Paris 1 – Panthéon – Sorbonne, Centre de Recherche en Informatique, 90 rue de Tolbiac, 75634 Paris cedex 13, France; E-mail: praknav@hotmail.in

Rim-Samia Kaabi, Université Paris 1 – Panthéon – Sorbonne, Centre de Recherche en Informatique, 90 rue de Tolbiac, 75634 Paris cedex 13, France; E-mail: rkaabi@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 organisational 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 organisations. 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 MAP and illustrates its use in an excerpt of a real case in the Electricity Supply Industry.

1. INTRODUCTION

Variability as a concept has proved to be useful in various engineering domains in which companies are not anymore faced to the development of a single product but to engineering product lines and families. The former represent the evolution of a given product, for example, a DVD player, whereas the latter integrates different product lines having commonality, such as DVD and MP3 player lines. 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 (Ommering, 2002; Tomphson, 2001) and,
b. adaptation of products to different customers and various organisational settings (Svahnberg, 2001).

Seeing the duality that exists between products and processes, our position is that business process families do exist today in companies and could beneficially be handled by introducing the concept of variability. Indeed, since the process wave initiated by Hammer and Champy (Hammer, 1994), large portfolios of business process models have been created and continue to develop as business process models are recognised as indispensable artefacts to driving business management and evolution. Besides, 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 is not just the replacement of one process by another. Rather process change management 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. As a matter of fact, one can recognize business process lines and families in current organizations today.

The foregoing suggests a move away from management of individual process to managing a set of similar processes considered as a whole, a family. Our proposal is 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 three sections. The next section introduces the MAP formalism and shows how it captures business feature variability. Section 3 presents an example and section 4 considers the adaptation of a business process model within a family.

2. CAPTURING BUSINESS VARIABILITY IN MAPS

We use the MAP formalism (Rolland, 2000) to capture variability of business processes modelled in an intentional manner.

2.1 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 an example in Fig. 6) with goals as nodes and strategies as edges between goals. The directed nature of the graph shows which goals can follow which one.

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 Fig.7). This leads to intentional process modeling as a hierarchy of maps.

As process models, maps can be compared to the various types of process modelling languages and formalisms that 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 (Ould, 1995)), state-based languages (e.g. UML state charts), an intention-oriented languages (e.g. Maps). The concept of goal is central in business process modelling 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" (Hammer, 1994). However, most process modelling 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 externalising *what* the process is intended to accomplish in the goal (Dietz, 2004). In contrast, intention-oriented process modelling 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 modelling such as MAP, follows the human intention of achieving a goal as a force, which drives the process. 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.

2.2 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 Fig. 1).

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 (Kang, 1990) for example, a feature is defined as *"A prominent or distinctive user-visible aspect, quality or characteristic of a software system or systems"*. In (Bosch, 2001), 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 Fig. 2. It shows through the strategies the different flows of activities provided to obtain the same result.

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

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. Fig. 3 shows an example of a bundle relationship.

Figure 1. A section

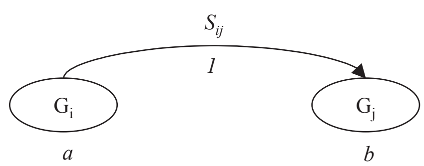


Figure 2. A multi-thread relationship

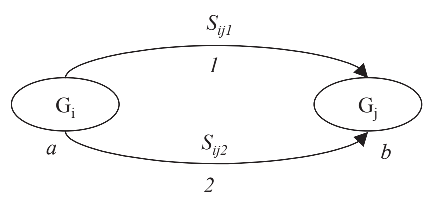


Figure 3. A bundle relationship

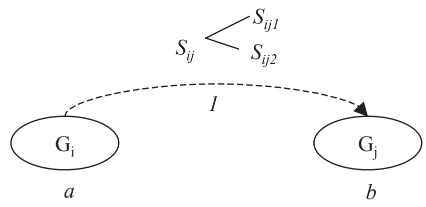
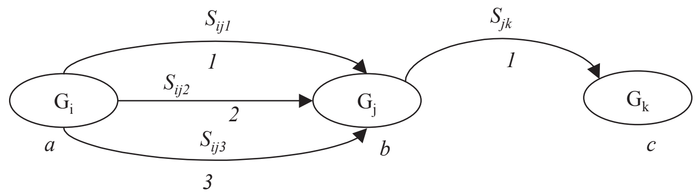


Figure 4. A path relationship



The bundle relationship expresses feature variability by grouping alternative features that are mutually exclusive.

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.

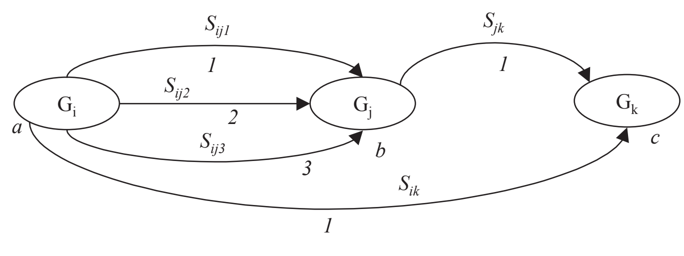
Fig. 4 represents a path relationship between the goals G_i and G_k, denoted respectively by a and c, which is composed of the multi-thread relationship containing the features ab_1, ab_2, ab_3 and the feature denoted bc_1. It expresses that in order to achieve the goal G_k, we must first select and execute one or many features among ab_1, ab_2 or ab_3 and then execute the feature bc1.

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 Fig. 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.

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.*

Figure 5. A multi-path relationship



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*.

2.3 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 MacNaughton and Yamada's algorithm (MacNaughton, 1960) 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 (" "), 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 :

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

In this paper we 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: $MT_{kl} = \{kl_1 \; kl_2 \; ...kl_n\}$ where the kl_i are the features related by the multi-thread relationship. Thus, the multi-thread represented in Fig. 2 is: $MT_{ab} = \{ab_1 \;\; ab_2\}$.

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 Fig. 3, 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 Fig. 4 is denoted: $P_{a,\{b\},c} = MT_{ab} . bc_l$.

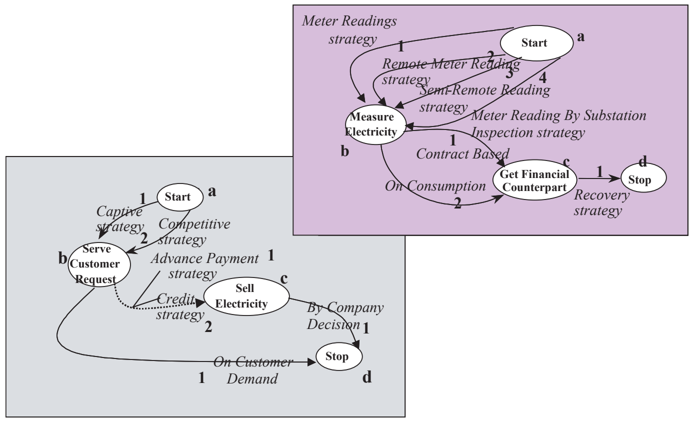
Multi-path relationship between two goals k and l is denoted $MP_{k,Q,l}$ where *Q* designates the set of intermediary goals used to achieve the target goal *l* from the source one *k*. A Multi-path relationship is based on the union operator "$\cup$" between alternative paths. Thus, the multi-path of Fig. 5 is denoted: $MP_{a,\{b\},c} = P_{ac} \quad MT_{ab} . P_{bc}$

In section 3, 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, namely *multi-path, path, thread* and *bundle* 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.

Figure 6. The ESM map sample

Figure 7 shows a refined view of the section bc_2 of this map itself expressed as a map



3. AN EXAMPLE

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 (Grosz, 1998). The map of Fig. 6 provides the top level intentional view of the electricity supply business family to support Electricity Supply Management (ESM).

The map shown in Fig. 6 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 energise 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.

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 ab_1 and ab_2 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

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	$P_{a,\{b,c\},d} = MT_{ab} . MP_{b\{c\}d}$ $P_{b,\{c\},d} = B^*_{bc} . cd_1$
Multi-Path	$MP_{b\{c\}d} = bd_1 \cup P_{b,\{c\},d}$
Bundle	$B_{bc} = bc_1 \otimes bc_2$
Multi-thread	$MT_{ab} = ab_1 \vee ab_2$

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 2. The initial formula generating all the paths between the goals *a* and *d* is: $Y_{a\{a, b, c,, d\},d} = (X_{a, \{b,c,d\}, a})^* . X_{a,\{b,c\},d} . X^*_{d,\{b,c\},d}$. The identified paths (and therefore composition of features) are summarized in Table 1.

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

4. 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 Fig. 7. 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 bc_2 of the map of Fig. 6, 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 Fig. 7. Each feature selection has however a payoff that can be analysed in the view of its combination to another one. The pay-off analysis for bc_2 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. The table shows that remote readings are a cost effective way to handle electricity measurement in both cases. Indeed, it is real time and therefore 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 bc_2 is then to keep the features ab_2 and ab_4 along with bc_1 and bc_2.

5. CONCLUSION

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 leader to the choices that are relevant to the satisfaction of her goals in terms of the properties of the business and there is no need to deal with technical configuration details.

This paper has reported the current status of on-going work. Future work consists of (a) implementing a configuration tool to adapt a business process model of a family using the map formalism and (b) developing a software tool to support navigation in a map to select dynamically the feature most appropriate to the situation at hand.

6. REFERENCES

Bosch, J., Florijn, G., Greefhorst, D., Kuusela, J., Obbink, H. & Pohl, K. (2001). 'Variability issues in Software Product Lines'. *Proceedings of the 4th International Workshop on Product Family Engineering (PEE-4)*, Bilbao, Spain.

Dietz, J.L.G. (2004). 'Basic Notions Regarding Business Processes and Supporting Information Systems'. *Proceedings of BPMDS'04, CAISE'04 Workshops Proceedings, Latvia, Riga*, (pp.160-168).

Grosz, G., Loucopoulos, P., Rolland, C. & Nurcan, S. (1998). 'A Framework for Generic Patterns Dedicated to the Management of Change in the Electricity Supply Industry'. *Proceedings of the 9th International Conference and Workshop on Database and Expert Systems Applications, DEXA '98*.

Table 2. Pay-off summary

<table>
<tr><td></td><td></td><td colspan="2">Get financial counterpart</td></tr>
<tr><td></td><td></td><td>Contract based</td><td>On consumption</td></tr>
<tr><td rowspan="4">Measure electricity consumption

I.

II.

III.</td><td>Meter reading by meter reader</td><td>Can be envisaged at sustainable cost if visits are achieved at a low frequency e.g. once or twice a year)</td><td>Excluded because too difficult to organise all visits at the required pace.</td></tr>
<tr><td>Remote reading</td><td colspan="2">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.</td></tr>
<tr><td>Semi-remote reading</td><td>Cost effectiveness is a linear function of the number of contracts per cluster of semi-remote reader.</td><td>Very costly if the number of customers paying on consumption, per cluster of remote reader is low.</td></tr>
<tr><td>Substation inspection</td><td>Only possible if the connected meter readers relate to single contract. Otherwise, calls for individual reading.</td><td>Cost effective way to handle the verification of consumers invoiced by remote reading clustered on the same substation.</td></tr>
</table>

Hammer, M., & Champy, J. (1994). 'Reengineering the Corporation-A manifesto for Business Revolution', Harperbusiness, (pp. 233).

Kang, K., Cohen, S., Hess, J., Novak, W. & Peterson, A. (1990). 'Feature-Oriented Domain analysis (FODA) Feasibility Study'. *Technical report CMU/SEI-90-TR-21*, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University.

MacNaugthon, R. & Yamada. (1960). 'Regular expressions and state graphs for automata'. *IEEE Transactions on Electronic Computers*, EC-9, (pp. 39-47).

Ommering, R.V. (2002). 'Building product populations with software components'. *Proceedings of the 24th International Conference on Software Engineering*, Orlando, Florida.

Ould, M.A. (1995). 'Business Processes: Modeling and Analysis for Reengineering and Improvement', *John Wiley & Sons*, may, (pp. 224).

Rolland, C. & Prakash, N. (2000). 'Bridging the gap between Organizational needs and ERP functionality'. *Requirements Engineering Journal*.

Svahnberg, M., Van Gurp, J., & Bosch, J. (2001). 'On the notion of variability in Software Product Lines'. *Proceedings of the Working IEEE/IFIP Conference on Software architecture*.

Thomphson, J., & Heimdalh, M.P. (2001). 'Extending the Product Family Approach to support n-Dimensional and Hierarchical Product Lines'. *Proceedings of the 5th IEEE Symposium on Requirements Engineering*, Toronto, Canada.