HAL

archives-ouvertes.fr

Matching ERP System Functionality to Customer Requirements

Colette Rolland, Naveen Prakash

HAL Id: hal-00707080

https://hal.archives-ouvertes.fr/hal-00707080

Submitted on 14 Jun 2012

Matching ERP System Functionality to Customer Requirements

Colette Rolland*, Naveen Prakash**

Université Paris1 Panthéon Sorbonne
CRI, 90 rue de Tolbiac,
75013 Paris, France
rolland@univ-paris1.fr

** British Council
17 Kasturba Gandhi Marg,
110001 New Delhi, India
praknav@hotmail.in

Abstract

Although procuring Enterprise Resource Planning systems from commercial suppliers is becoming increasingly popular in our industry, fitting those systems to customer requirements remains problematic. In this paper, we propose an approach for matching ERP system functionality to customer requirements. The assumption made is that the ERP system postulates a set of requirements that are worth eliciting from the ERP documentation as abstractions of the ERP system functionality. Then, the requirements engineering process is a process that matches the ERP set of requirements against organisational ones. Those requirements that match, perhaps after adaptation identify the ERP system features and their adaptations, that must be included in the ERP installation. To facilitate the matching process, the ERP requirements and the organisational ones are both expressed using the same representation system, that of a Map. The paper presents the Map representation system and the matching process. The process is illustrated by considering the Treasury module of SAP and its installation in the financial management of a cultural exchanges unit.

1. Introduction

Although purchasing commercial off-the-shelf (COTS) products such as Enterprise Resources Planning (ERP) systems is increasingly popular in our industry, fitting these systems to customer requirements remains problematic. This requirements 'fitting' must be done at two different levels:
1. the COTS/ERP system with the best fit is to be selected and acquired,
2. once an ERP system has been procured then, it must be customised to carry out the specific task at hand.

Considerable attention has been given to (1) above and includes requirements acquisition for COTS software selection [11], process guidance to enable evaluation of COTS software [12], [4], employing use cases in a COTS tender [10], and the procurement process [15]. Other factors considered are functional capabilities, interfaces, architectural issues [7] and interoperability with legacy systems [5] as well as review of important procurement factors and research ideas [6] and the need for a better definition of requirements for off-the-shelf software [13].

In contrast, the customising activity of (2) above has been looked upon mainly by ERP system vendors and software houses that are accredited for ERP system installation. For instance, SAP has developed the ASAP method to support the deployment of SAP/R3 in an organisation. A number of practice surveys show that customisation is a detailed process which is difficult to perform and is one of the main causes of failures of ERP installations. This finds support, for example, in the benchmarking done by the Partners Group in 1998, where one of the drawbacks found in ERP installation was the difficulty of alignment to specific requirements of the enterprise. In this paper, we attempt to mitigate this difficulty by developing an *approach to align ERP system functionality to organisation requirements.* In order to emphasise the requirements orientation of the approach we call it the *ERP requirements driven approach.*

In its traditional practice, requirements engineering acquires, models and validates functional and non functional requirements used to develop a computer based solution meeting these requirements. The assumption is that systems are built ab initio from the set of requirements. In contrast, ERP systems are available off the shelf. Therefore, requirements engineering for ERP installation must take into account the implications of this fact. These are:
- a generic software solution has already been selected by the organisation and is embodied in the ERP system. To minimise costs and risks, the required ERP installation must be confined within the ERP

offer and the deviation from the offer must be minimised,

- the success of an ERP installation is critically dependent on the customising process. It is only through the parameterisation performed that the strategic decisions of an organisation are reflected [9] in the ERP installation. Thus, besides speed and cost of customisation, appropriate parameterisation is crucial,
- a number of organisational requirements have been elicited already to support the decision of acquiring this ERP system,
- the capability of the ERP system is described in terms of its functionality and its range of variants. Clearly this functionality subsumes a set of requirements even though these requirements are not explicitly stated in ERP system documentation.

Our experience with installation of ERP systems [1][9] is that there is a language mismatch between ERP experts and organisation stakeholders because the customisation process focuses on the functionality of the ERP system. This mismatch exposes the ERP system installation to the danger of failing to meet the requirements of organisations. Our proposal is therefore to move from *functional alignment* of the ERP installation to *requirements alignment* thereby causing the customisation process to focus on requirements rather than on functionality.

In order to achieve this, our proposal is to develop the *ERP requirements driven approach* which consists of

- abstracting from the existing ERP functionality,. the set of requirements that it subsumes. This will establish the link between ERP system requirements and ERP system functionality,
- performing the alignment at the requirements level,
- deriving from the aligned requirements the adaptations and extensions of the selected functionality by using the ERP requirements to functionality link

In this paper, we will deal with the first two aspects of this approach. Our *ERP requirements driven process* is depicted in Figure 1. As shown, the first step consists of abstracting from the ERP system functionality, the set of requirements it subsumes.

The second step is the matching process to understand which of the ERP set of requirements match the requirements of the organisation. The result of this step, the end product of the process, is the set of requirements of the ERP based system that will be installed. This product may be a subset of the ERP set of requirements, perhaps suitably augmented with those that are not contained in the ERP set. Even though this process is

principally driven by the ERP set, it is clearly necessary to balance it with organisational requirements (Fig.1). As widely accepted [8], we view organisational requirements as the pair of As-Is and To-Be requirements. As-Is requirements are those that are currently satisfied in the functioning of the organisation whereas To-Be requirements are the ones to be satisfied in its future functioning.
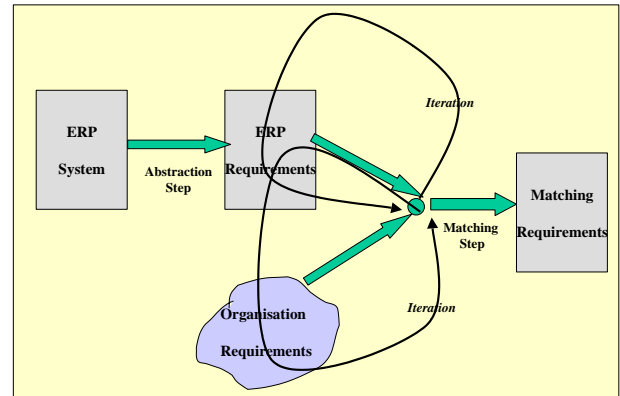

Figure 1: The ERP requirements driven process

The matching process is an iterative one (see Fig.1). This allows

- The balancing of the ERP drive with the To-Be and As-Is drives
- The progressive refinement of the product at each iteration
- The process to be controlled to reach the appropriate level of requirements understanding.

At each iteration of the process, only those requirements that have not yet been matched are investigated and detailed. This is in line with the fact recognised by Anthony Finkelstein and colleagues [6] that complete requirements specification is not always needed in software package requirements engineering.

To support the matching process we need a representation system for the As-Is, To-Be, ERP set of requirements, and the set of matching requirements. We choose to use the same uniform representation, that of a Map [14], for all of these. An overview of the Map is provided in section 2 of this paper.

This paper presents the *ERP requirements driven process*. It illustrates the process by considering the Treasury module of SAP R/3 and its installation in the financial management of the cultural exchanges unit of a diplomatic mission. It is organised as follows. In the next section, we provide an overview of the notion of a map. Section 3 presents the ERP requirements driven process. The case study is introduced in section 4 and the maps of

SAP R/3 Treasury module, the As-IS and To-Be maps of the Unit are discussed. The matching process is applied to the case study to obtain the Matched map in section 5. Section 6 is the concluding section.

2. Overview of a Map

In this section we introduce the key concepts of a map. The aim is to lay a basis for the maps that will be constructed to illustrate the *ERP requirements driven process.* A detailed description of the notion of a map can be found in [14], [3].

The key concepts of the map and their inter-relationships are shown in the map meta-model of Figure 2 using standard UML notations. These are as follows:

- A *map* is composed of one or more sections. A section is an aggregation of two kinds of intentions, the source and target intentions together with a strategy.
- An *intention* is a goal that can be achieved by the performance of a process. For example, *Make Room Booking* is an intention to make a reservation for a room in a hotel. Similarly, *Accept Payment* is another intention. We postulate that each map has two special intentions, *Start* and *Stop*, to begin and end the map respectively.
- A *strategy* is an approach, a manner to achieve an intention. In our example, let it be required that bookings can be made on the internet. This is a way of achieving our room booking intention: it is a strategy.

Note: As discussed in the Introduction, we treat a goal together with the way it can be achieved as a requirement. Thus, *Make Room Booking by the Internet strategy* is a requirement that has to be met in our room booking system.

- A *section* is an aggregation of the source intention, the target intention, and a strategy. A section expresses the strategy using which, starting from a source intention, the target intention can be achieved. For example, the aggregation of the source intention *Start*, the target intention *Make Room Booking* and the *Internet strategy* defines a section *<Start, Make Room Booking, Internet strategy>*. Here, the *Internet strategy* characterises the flow from the source intention *Start* to the target intention *Make Room Booking* and the way the target can be achieved.

- *Path relationship*: this establishes a precedence/succedence relationship between sections. For a section to succeed another, its source intention must be the target intention of the preceding

one. In our room booking example, in addition to the requirement to *Make Room Booking by the Internet strategy* let there be another requirement to *Accept Payment by Electronic Transfer strategy* Evidently, this requirement can be fulfilled after a booking has, in fact, been made. Thus, there is a section *<Make Room Booking, Make Payment, Electronic Transfer strategy>*. Then there is a path relationship between this section and the earlier section, *<Start, Make Room Booking, Internet strategy>*: *Make Room Booking* is the target of one and the source of the other section.
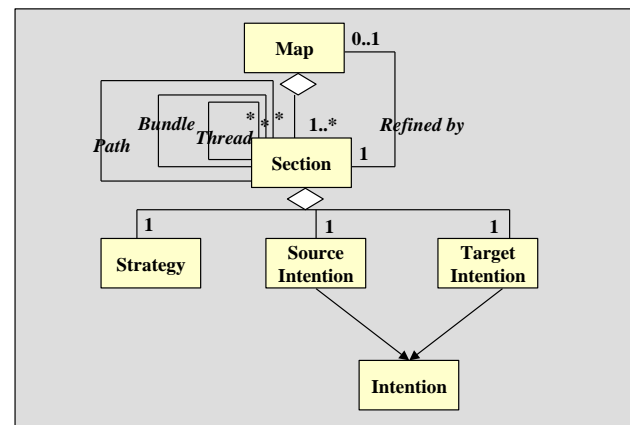

Figure 2: The map meta-model

- *Thread relationship*: It is possible for a target intention to be achieved from a source intention in many different ways. One or more of these sections can be used in realising the target intention. Each of these ways is expressed as a section in the map. Such a map topology is called a *multi-thread* and the sections participating in the multi-thread are said to be in a *thread relationship* with one another. For example, let it be required that payment can be accepted by both electronic means and by credit card. Thus we have two requirements, *Accept Payment by Electronic Transfer* and *Accept Payment by Credit Card* both of which (by an extension of the discussion above) have the same source intention, *Make Room Booking* and the same target intention *Accept Payment.* Then the two sections *<Make Room Booking, Accept Payment, Electronic Transfer strategy>* and *<Make Room Booking, Accept Payment, Credit Card strategy>* are in a thread relationship with one another because they represent two different ways of achieving *Accept Payment* from *Make Room Booking*.
- *Bundle relationship*: A section that is a bundle of other sections, is an expression of the fact that exactly one of its sections can be used in realising the target intention. Consider *Make Room Booking* and

Accept Payment once again. Let it be that the hotel has entered into an agreement with an airlines to provide rooms against miles earned by passengers. In such a case, payment is accepted either normally or from the airlines and the two cannot be done together. Then, we can consider *<Make Room Booking, Accept Payment, Payment strategy>* to be a bundle of the two sections, *<Make Room Booking, Accept Payment, Normal Payment strategy>* and *<Make Room Booking, Accept Payment, Airline Miles strategy>*.

- *Refinement relationship*: Figure 2 also shows that a section of a map can be refined as another map through the *refinement relationship*. 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. Refined maps of a section will be explicitly pointed out in the paper.

We will represent each map as a directed graph from *Start* to *Stop*. In this graph, intentions are represented as nodes and strategies as edges between these. The graph is directed because the strategy shows the flow from the source to the target intention. As an example consider the map of Figure 3 which contains six sections MS0 to MS5.
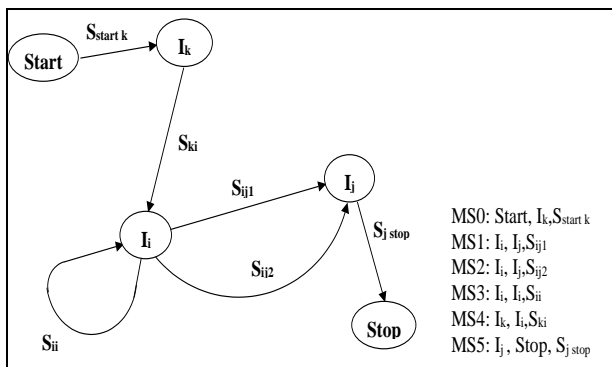

Figure 3: The map as a graph

MS1 and MS2 constitute a multi-thread. There are multi-paths from Start to Stop : MS0, MS4, MS3, MS1, MS5 is one example; MS0, MS4, MS2, MS5 is another one.

3. The ERP Requirements Driven Process

The aim of the *ERP requirements driven process* is to elicit the requirements for the ERP system installation. These requirements are expressed in the *Matched Map.* The assumption of the process is that this map is obtained by matching the requirements that can be fulfilled by the

ERP system with the specific requirements of the organisation. To facilitate matching, both these requirements are expressed as maps. The former are expressed as an *ERP map* whereas the latter is the pair of *As-Is* and *To-Be maps.* In conformity with the accepted RE view [8], the As-Is map represents requirements that are currently fulfilled whereas the To-Be map projects the new requirements.

The *ERP requirements driven process model* is shown in Figure 4. The key process intentions are *Construct As-Is, To-Be, ERP maps* and *Construct Matched Map*. They delineate three steps in the process. The first step leads to the construction of the As-Is, To-Be, and ERP maps based on an abstraction activity. The second step is the matching sub-process itself. It can be taken forward by three different drives, the ERP, To-Be and As-Is drives. Additionally, the matching process allows iteration as shown in the Figure by the Feed-back edge originating from *Construct Matched Map*. Finally, the third step verifies the accuracy of the Matched Map.
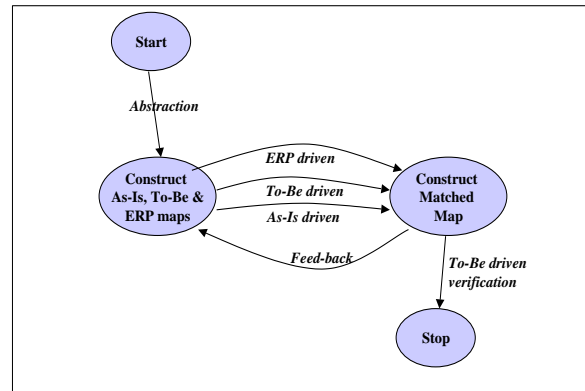

Figure 4 : The ERP requirements driven process model

We now describe the three steps in turn. First, we have the *abstraction step.* Here,

- The *As-Is map* results by abstracting from current practice to describe the currently achieved goals/requirements. It is useful for critiquing the current situation and thereby identifying customer requirements for the *To-Be map*. Additionally, it also serves as a reference to estimate the deviation between the new solution and current practice.
- The *To-Be* map reflects the goals/requirements that the organisation would like to satisfy by installing the ERP system.
- The *ERP map* specifies the goals/requirements that the ERP system supports.

The second step, *the matching step*, is the heart of the process. It produces the *Matched Map* which is the output of the requirements process and the input to the

installation process. Most of the intentions and strategies of the Matched Map are obtained from the ERP map and match the To-Be requirements. Others may not be available in the ERP map and will require in-house development. In such a case, the Matched Map helps in their identification. On the contrary, all the intentions and strategies of the ERP map may not be included in the Matched Map. This corresponds to the ERP functionality that is not matching the requirements in the To-Be map.

The construction of the Matched Map can be by the *ERP-driven, As-Is driven and To-Be driven strategies*. Each of these consider the intention and strategy of the corresponding map from Start to Stop in order to decide if they (a) match the requirements exactly and so must be included in the Matched map, (b) need adaptation before their inclusion in the Matched Map, and (c) are irrelevant.

We expect the *ERP-driven strategy* to be the most frequently used. This is because the raison d'être of ERP systems is their ability to capture almost all requirements of back office information systems. The ERP-driven strategy provides an open-ended solution space within which a range of solutions meeting the requirements can be elicited. In contrast, the *As-Is* and *To-Be driven strategies*, in so far as they proceed from a specified targeted requirement, circumscribe the search space to meeting the target requirement.

The matching sub-process is a proactive process that may need refinement of the *As-Is*, *To-Be*, or the *ERP-map* at any moment in the construction of the *Matched Map*. This can be achieved by following the *Feed-back strategy* of Figure 4. For example, a section of the *ERP map* may be adequate to make the decision of its relevance and inclusion in the *Matched Map* without an analysis of its refinement whereas this analysis might be required at all levels of refinement provided by the map.

Finally, the third step to *Stop* the process, checks that the *Matched Map* meets all the requirements of the *To-Be map*. It determines the accuracy of the *Matched Map* and terminates the process.

4. The Case Study: Building Maps

In this section we first provide an overview of SAP R/3 and its Treasury module before presenting the Treasury map. We then present and discuss the *As-Is* and *To-Be maps*.

4.1 SAP Overview

In this section we provide an overview of SAP R/3 functionality and outline its treasury module. This information has been obtained from Using SAP R/3 [2].

The treasury module will be used in the next sub-section to build a treasury map.

Overview
SAP R/3 aims to provide a complete infrastructure for corporate information processing. It does this by offering a set of standard business applications. These can be tailored to meet the specific needs of an organisation.

The set of standard applications consists of about 20 *modules* like Treasury, Production Planning, Sales and Distribution, Financial Accounting etc. Each module consists of a number of *components*. For example, the Treasury module has 3 components, Cash Management, Funds Management, and Treasury Management. Each component has a number of *functions*. Thus, the Cash Management component of the Treasury module provides, among others, functions for managing cash accounts and managing clearing accounts.

Modules of R/3 do not work in isolation but may be *inter-related* to one another. For example, the Treasury (TR) and the Investment Management (IM) modules overlap because the functions of IM-FI Financial Investments are the same as those of TR-TM Treasury Management. In addition, the TR-Treasury module is related to the FI-Financial Accounting module as it provides some enhanced functions additional to those in FI-IM Investment Management and FI-AM Fixed Assets Management.

Some modules, components of modules, and functions of components are *optional*. Further, SAP provides, through ABAP/4, a means to introduce new functionality to meet any situations that it cannot handle. As a result of this flexibility, each R/3 installation can be built to fit the client organisation. The process of adapting SAP R/3 to an organisation is called *customising*. It consists of selecting the components and fine tuning their functions. Customising is facilitated by a set of tools available as part of R/3 BASIS which is the core of SAP R/3. Thus, R/3 Analyser, a constituent of R/3 BASIS, is used to select the functions of interest from the R/3 Reference Model. Fine-tuning is mainly concerned with the data to be manipulated and the transactions to manipulate them. For example, the accounting period can be the fiscal year or a shorter period tuned to the practice of the company.

Description of the Treasury Module

The TR-Treasury module includes three components :
1. TR-CM Cash Management for the day to day management of short-term and long-term cash flows. It provides planning capabilities to ensure that fund reservations are not likely to compromise future liquidity.

2. TR-FM Funds Management for providing comprehensive support to prepare budgets and their assignment to budget commitment fund centres. Different versions of the budget can be maintained separately in budget data objects.
3. TR-TM Treasury Management supports business transactions from the trading stage to back-office processing and then to eventual posting in the financial accounting system.

Each component contains functions. For example, the TR-CM Cash Management component has the following functions :

- Managing cash accounts and clearing accounts
- Forecasting medium-term cash flow
- Making an annual cash-flow plan
- Arranging automatic bank account clearing
- Using electronic banking facilities.

4.2 The Treasury Map

The overall objective of the Treasury module is medium and long-term financial planning along with medium-term management and control of revenues and expenditures. The focus is on providing an accurate view of the financial state of the company in the short and medium term future. The former is provided by cash and treasury management functionality while the latter is done by the funds management functionality.

This is reflected in the Treasury map shown in Figure 5 by the two intentions, *Manage Liquidity* and *Plan Funds* respectively. *Manage Liquidity* can be done once the medium term financial budget has been established by the *Plan Funds* intention. This justifies the precedence of the intention, *Plan Funds* over the intention, *Manage Liquidity* of Figure 5.
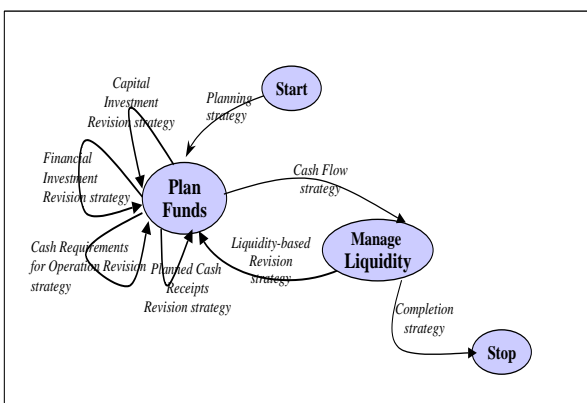

Figure 5 : The Treasury Map

The section *<Start, Plan Funds, Planning strategy>* captures the different ways in which the total budget can

be built and assigned to different budget commitment fund centres. The refinement of this section to bring this out is in the refined map of Figure 6. The map consists of three intentions *Forecast Budget Item*, *Construct Budget*, and *Assign to Budget Centres*. The achievement of *Forecast Budget Item* leads to the valuation of items comprising the budget. The *Construct Budget* intention builds the total budget once the individual items have been accounted for. This explains the ordering between *Forecast Budget Item* and *Construct Budget* shown in the Figure. The third intention, *Assign to Budget Centres,* assigns budgets to the different budget commitment funds centres. Again, this intention can be achieved only after the budget has been constructed and so follows *Construct Budget* in the Figure. The flow from *Start* to *Stop* in the refined map can be seen to reflect a plan to achieve the intention *Plan Funds* of the main map.
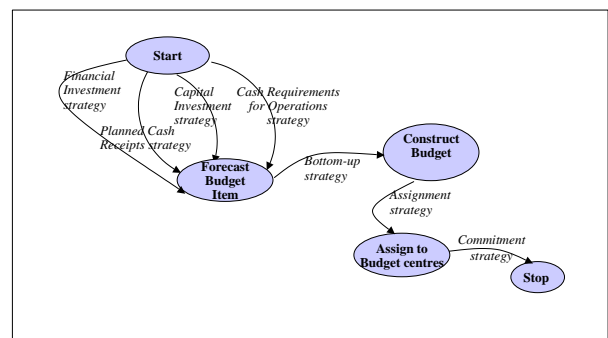

Figure 6: The Refined Map of section *<Start, Plan Funds, Planning strategy>*

Whereas there is only one possible path from *Forecast Budget Item* to *Stop*, the refined map shows that there are four strategies to *Forecast Budget Item* from *Start*. These constitute a multi-thread between these two intentions. The strategies are, *Financial Investment strategy, Planned Cash Receipts strategy, Capital Investment strategy, and Cash Requirements for Operations strategy*. The *Financial Investment strategy* provides a way of forecasting the three main financial asset items of a company, viz., securities, loans, and time deposits. The *Planned Cash Receipts strategy* is used to forecast the income receipt items that are expected to accrue over the plan period. The *Capital Investment strategy* is in fact a bundle of two strategies, the *Flat Rate Investment strategy* and the *Capital Investment Program strategy*. The latter provides for both annual investment as well as continuous, rolling, investment over a number of years on capital goods. The *Flat Rate Investment strategy* is used to provide funds for asset items that are to be capitalised directly. It is used as an alternative to the *Capital Investment Program strategy*. Finally, the *Cash Requirements for Operations strategy* gives a way of budgeting expenditure items over the business activities that are planned to be carried out.

This case of a multi-thread shows how the map displays different strategies made available by the SAP system to fulfil a given intention. Recall that in a multi-thread (section 2), strategies are not exclusive and one or more of these can be used in combination or separately. Our case (Figure 6) provides a range of strategies that can be used by organisations to forecast their budget items. Some of these may be applicable to an organisation and some others may not be. For example, an organisation that is either not allowed to invest in financial assets or does not do so, will not use the *Financial Investment strategy*. Similarly, a charitable organisation may not have any *Planned Cash Receipts* and so will not deploy this strategy.

With the achievement of the *Forecast Budget Item* intention of Figure 6, the total budget has now to be built. In other words, the *Construct Budget* intention has to be fulfilled. The map shows that this can be done by using the *Bottom-up strategy.* Through this strategy, items valuated separately are combined to form the total budget. This strategy allows budget allocation to subordinate items even if the higher-level budgets have not yet been allocated.

Once the budget has been constructed, sections of it have to be assigned to budget commitment centres. This justifies the linear order (see Figure 6) *between Construct Budget* and *Assign to Budget Centres*. Finally, the *Commitment strategy* is used to obtain approvals and signals the achievement of the *Plan Funds* intention of the main map (Figure 5). This reflects the best estimate of the costs and revenues that should be realised in the plan period.

Let us then proceed with the main map of Figure 5. As the period passes, fresh targets can arise as a result of operating costs, investments, cash receipts etc. that were not anticipated exactly in the plan. The decision to revise the plan can be made and this revision can be performed using the four revision strategies shown in the map of Figure 5. These are the *Capital Investment Revision strategy, Financial Investment Revision strategy, Cash Requirements for Operations Revision strategy,* and *Planned Cash Receipts Revision strategy.* These are different from the corresponding strategies for *Forecast Budget Items* in Figure 6 in that they support revision and not creation of the items.

As mentioned earlier, once the *Plan Funds* intention has been achieved the *section <Plan Funds, Manage Liquidity, Cash Flow strategy>* can be triggered. The aim behind this section is to provide means for maintaining liquidity in order to fulfil payment obligations. This includes medium term financial planning in the form of

annual cash plans and the monitoring of all transactions that have payment advice notes, control cash and invest cash. The achievement of *Manage Liquidity* by the *Cash Flow strategy* is a complex task that is refined into the map shown in Figure 7.
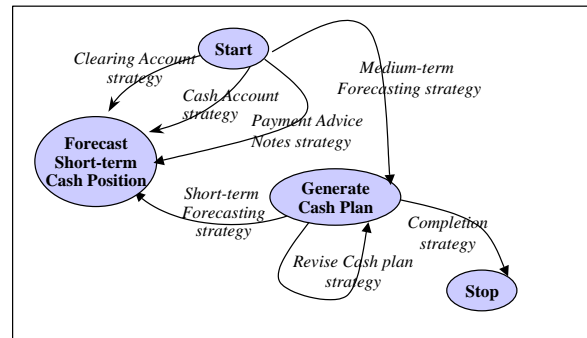

Figure 7 : Refinement of the <Plan Funds, Manage Liquidity, Cash Flow strategy> section

In this map, the annual cash plan results when the *Medium-term Forecasting* strategy is used to *Generate Cash Plan.* The annual cash plan shows how liquidity is secured over the period by exercising financial control. The *Medium-term Forecasting strategy* allows to set the annual plan at any level of detail and for any arrangement of the organisational structure. This is shown by the section *<Start, Generate Cash Plan, Medium-term Forecasting strategy>* in Figure 7.

The annual cash plan is complemented by a short-term cash plan which forecasts liquidity in the short-term period. This period can extend from several days to a week or even a month. The production of this plan results from a path between *Start* to *Forecast Short-term Cash Position* through *Generate Cash Plan.* The achievement of the *Forecast Short-term Cash Position* intention forecasts the cash position predicted by the three strategies, *Clearing Account strategy, Cash Account strategy, or Payment Advice Note strategy* respectively. These three exploit transaction information made available in Clearing accounts, Cash accounts and Payment Advice Notes. Based on the forecasted cash positions, *the Short-term Forecasting strategy* is deployed to generate the short-term cash plan. This corresponds to the section *<Generate Cash Plan, Forecast Short-term Position, Short-term Forecasting strategy>.*

Since deviations may occur over time, the annual cash plan can be modified to reflect these. This is modelled by the section *<Generate Cash Plan, Generate Cash Plan, Revise Cash Plan strategy>.*

Finally, the use of the *Completion strategy* ensures that at the end of the plan period the appropriate documentation

is generated. This signals the complete fulfilment of the intention *Manage Liquidity* (Figure 5) for the ending plan period and triggers a smooth roll-over to the next plan period to *Manage Liquidity*.

The main map of Figure 5 contains the section *<Manage Liquidity, Plan Funds, Liquidity-based Revision strategy>* which reflects the fact that the cash position may have an effect on the budgets of the budget funds commitment centres.

The *<Manage Liquidity, Stop, Completion strategy>* section represents the process of closing the budgeting activities for the plan period.

4.3 The As-Is Map

The *ERP requirements driven process* is applied to the finance management of the cultural exchanges unit of a diplomatic mission. This Unit obtains funding directly from its country and generates some local income through its Library and Language Education Centres, and by raising sponsorships. Its expenditures, besides those of infrastructure, staff salaries etc., are on the planned cultural activities and on funding exchange programs between the countries. The Unit focuses on controlling the expenditure side so that it does not exceed the grants-in-aid received from its country. The *As-Is map* reflecting the key intentions and strategies used in this Unit is in Figure 8.
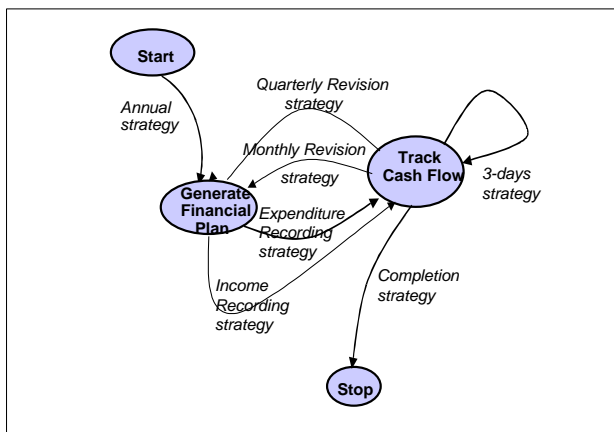

Figure 8 : The As-Is Map

The *As-Is Map* has two main intentions, *Generate Financial Plan* and *Track Cash Flow*. Since tracking can only be done after the plans have been generated, *Track Cash Flow* follows *Generate Financial Plan*. The financial management of the Unit is based on two plans, the annual budget of the Unit and the cash plan for the entire year projected on a month-by month basis. This is captured by the *section <Start, Generate Financial Plan,*

Annual strategy>. The *Generate Financial Plan* intention produces both the budget and the cash plan for the year using the Annual strategy.

With the two plans generated, *Track Cash Flow* aims to ensure the liquidity of the Unit. It does this by using the *Income Recording strategy* for keeping track of the amount of income and the *Expenditure Recording strategy* for keeping track of the amount of expenditure. Each of the sections, *<Generate Financial Plan, Track Cash Flow, Income Recording strategy>* and *<Generate Financial Plan, Track Cash Flow, Expenditure Recording strategy>* are triggered every day so as to keep track of the cash flow on a daily basis. The Unit has a policy that enough cash should be available to fund three days expenditure. This is because it takes three days to obtain its grant money from its country. This is modelled in the map by the reflexive section *<Track Cash Flow, Track Cash Flow, 3-days strategy>*.

Any deviations between the generated plans and the tracked cash flow is fed back to fine tune the plans. This is expressed in the map by the two sections, *<Track Cash Flow, Generate Financial Plan, Monthly Revision strategy>* and *<Track Cash Flow, Generate Financial Plan, Quarterly Revision strategy>*. The former allows the monthly projections made in the annual cash plan to be revised whereas the latter allows revision of the annual budget. The names of the two strategies indicate the periodicity of the revisions, monthly for the cash plan and quarterly for the annual plan.

Finally, the *Completion strategy* to *Stop* helps in producing the end-of-the-year statement on budget compliance and reports any deficits or surpluses that are generated by the Unit.

4.4 The To-Be Map

The cultural exchange Unit operates in a highly centralised mode and, given that it runs four regional offices in the foreign country, it aims to delegate financial planning responsibilities to its four offices. Whereas freedom to budget income and expenditure as well as to plan cash flow is to be provided, investments in financial assets are to be kept under control of the corporate office. Under this situation it is no longer possible to construct the annual budget and the cash flow together and it is only after the regional budgets have been negotiated and approved that the regional offices can plan their cash flows.

These changes motivate the *To-Be map* of Figure 9. The two intentions *Manage Budget* and *Manage Cash Plan* reflect the required separation between budget management and cash plan management respectively. In

order to reflect the temporal requirement of the Unit, the *Manage Cash Plan* intention follows the *Manage Budget* intention. The use of the *Medium-term Forecasting* strategy to flow from *Manage Budget* to *Manage Cash Plan* results in the annual cash plan being produced.
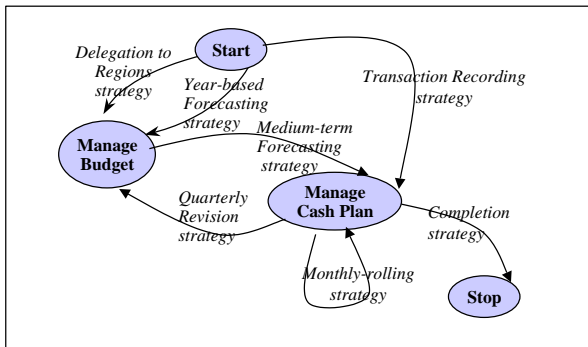

Figure 9 : The To-Be Map

The *Year-based Forecasting strategy* maintains the practice of preparing an annual budget. The *Delegation to Regions strategy* introduced in the map to *Manage Budget* from *Start* is a simple way to reflect the requirement for decentralisation. The currently used (see *As-Is map*) *Income Recording* strategy and *Expenditure Recording* strategy are still to be used in the *To-Be map* and are represented by the *Transaction Recording* strategy to *Manage Cash Plan*. The *Montly-rolling* strategy to revise the cash plan every month is similar to the *Monthly Revision* strategy of the *As-Is map*. The *Quarterly Revision* strategy and the *Completion* strategy in the *To-Be map* have the same meanings as the corresponding strategies in the *As-Is map*.

In accordance with the *ERP requirements driven process*, the *To-Be map* expresses a broad, high level view of the main changes in the requirements that are to be accommodated. It acts as a motivation for the production of the *Matched Map* by the matching-based requirements engineering process. However, the To-Be map is not a static document but can be progressively modified as the *Matched Map* is developed. The modifications that occur make the *To-Be map* more precise even though it continues to reflect requirements at a very high level.

5. The Case Study: Matching Maps

The first step of the *ERP requirements driven process* of Figure 4 is now complete since the *As-Is*, *To-Be*, and the SAP Map have been constructed. The next step in the process is the elicitation of the requirements for the installation of the SAP system that will be expressed in terms of the *Matched Map*. There are three strategies available for doing this and all these will be used simultaneously. However, the SAP driven strategy will be the main drive. The To-Be driven strategy will help in

discovering whether an element included in the *Matched Map* is matching the requirements or not. Finally, the As-Is driven strategy will provide background information to ensure a smooth transition between the old and the new system.

The construction of the *Matched Map* is illustrated by a step-by step process. In the following description the drive behind each step (Figure 4) is identified in parenthesis. The final result is presented in Figure 10.

Step 1 : Initiating the construction of the Matched Map (SAP driven)

The SAP map of Figure 5 proposes the starting section, *<Start, Plan Funds, Planning strategy>* to produce the annual budget and its allocation to budget centres. This matches the section *<Start, Manage Budget, Year-based Forecasting strategy>* of the *To-Be Map*. Therefore, it is decided to keep the intention *Plan Funds* and its associated strategy in the *Matched Map*.
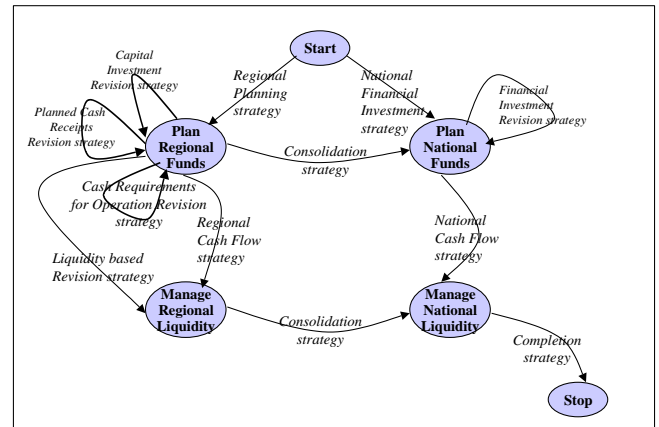

Figure 10 : The Matched Map

Step 2 : Dealing with the decentralisation requirement (To-Be driven)

In the *To-Be map*, the section, *<Start, Manage Budget, Delegate To Regions strategy>* calls for decentralisation of budget management to the regional offices. The Unit wishes to give autonomy to its four regions to manage their budgets. This includes planning for the operations that will be carried out in the region, the cash receipts and any capital investments to be made. However, regions are not allowed to do financial investments which is the sole responsibility of the national level. Besides, the responsibility of the national level is to develop a consolidated budget out of the regional budgets and obtain approval of its country. This calls for the splitting of *the <Start, Plan Funds, Planning strategy>* section suggested in step 1 above into two sections namely, *<Start, Plan Regional Funds, Regional Planning*

strategy> and *<Start, Plan National Funds, National Financial Investment strategy>* (see Figure 10).

Step 3: Refining the sections <Start, Plan Regional Funds, Regional Planning strategy> and <Start, Plan National Funds, National Financial Investment strategy> (SAP driven)

Since these two sections result from the splitting of the same original section in the SAP map, the refinement of the latter supports the refinement of both of these. The changes implied by the decentralisation of budget management require specific adaptations of the refined map of the SAP section (see Figure 6), *<Start, Plan Funds, Planning strategy>,* for each of the two sections of the Matched Map. These are shown in Figure 11 and Figure 12 respectively.
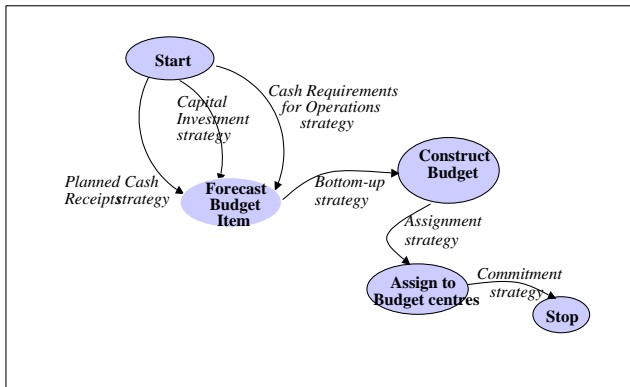

Figure 11: The refined map of the *<Start, Plan Regional Funds, Regional Planning strategy>* section of the Matched Map

The map in Figure 11 includes only three strategies to *Forecast Budget Item* namely, *Planned Cash Receipts strategy, Capital Investment strategy, and Cash Requirements for Operations strategy*. This is to account for the fact that the regional level is not concerned with Financial Investments. The rest of the map remains the same as in Figure 6.

The map in Figure 12 includes only one strategy, namely, the *Financial Investment strategy* and the rest of the map remains the same. This is because the National level is only planning financial investments. It is to be noted that the budget funds commitment centres are now the four regional offices.

Step 4: Completing the decentralisation requirement (To-Be driven)

In order to match the decentralisation requirement completely, it is necessary to model the fact that the *Plan National Funds* intention can only be achieved after *the*

Plan Regional Funds intention is achieved. As mentioned earlier, this is because the national level consolidates the regional budgets. This leads to the introduction of the *Consolidation strategy* in the *Matched Map* as a flow from the intention *Plan Regional Funds* to the intention *Plan National Funds* (Figure10).

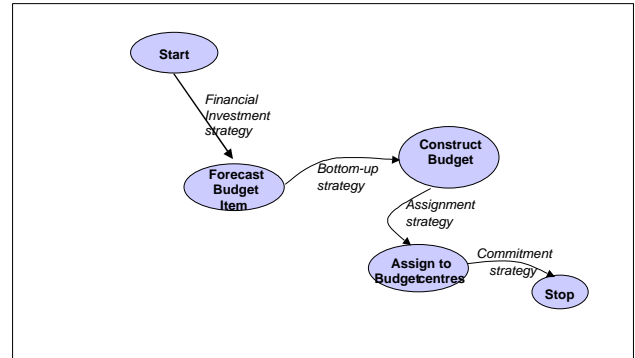

Figure 12 : The refined map of the *<Start, Plan National Funds, National Planning strategy>* section of the Matched Map

Step 5: Considering the budget revision strategies (SAP driven)

The SAP map provides for revision of the budget and assignment to funds commitment centres by the multi-thread of reflexive sections from *Plan Funds* to *Plan Funds* with the four revision strategies, *Capital Investment Revision strategy, Planned Cash Receipts Revision strategy, Cash Requirements for Operations Revision strategy, and Financial Investment Revision Strategy*. In accordance with the split of *Plan Regional Funds* and *Plan National Funds*, only the first three of these are used in *the Matched Map* to *Plan Regional Funds* whereas the last of these is used to *Plan National Funds* (Figure10).

Step 6: Flowing to liquidity management (SAP driven)

In the SAP map there is a flow from *Plan Funds* to *Manage Liquidity* based on the *Cash Flow strategy*. This obviously matches the requirement in the To-Be map that there is a flow from *Manage Budget* to *Manage Cash Plan* based on the *Medium-term Forecasting strategy*. Therefore, the *Manage Liquidity* and its associated *Cash Flow* strategy are introduced in the Matched Map.

Step 7: Effect of decentralisation (To-Be driven)

Now, even if the *To-Be map* does not explicitly express the requirement for decentralised management of liquidity, the policy of the Unit that the manager of the budget is also the manager of liquidity calls for the regions to manage their liquidity and the national level to manage its own. In other words, the *<Plan Funds,*

Manage Liquidity, Cash Flow strategy> section introduced in the *Matched Map* in step 6 is to be split up into two sections, *<Plan Regional Funds, Manage Regional Liquidity, Regional Cash Flow strategy>* and *<Plan National Funds, Manage National Liquidity, National Cash Flow strategy>*. These are included in the Matched Map of Figure 10.

Step 8: Understanding the sections <Plan Regional Funds, Manage Regional Liquidity, Regional Cash Flow strategy> and <Plan National Funds, Manage National Liquidity, National Cash Flow strategy> (SAP driven)

The SAP map proposes that the section *<Plan Funds, Manage Liquidity, Cash Flow strategy>* should be refined further into the map of Figure 7. This serves as a basis for the refinement of our two sections. As a consequence of the decentralisation requirement, it is apparent that the management of liquidity has to be performed differently at the regional and national levels. However, the *To-Be map* does not provide any specific requirement to handle it. Reference to the *As-Is map* only shows that in the existing practice management of liquidity is reduced to tracking the cash flow. Thus, the refinement is driven by the SAP refined map of Figure 7 in which necessary adjustments are made.

Step 9: Refining the section <Plan Regional Funds, Manage Regional Liquidity, Regional Cash Flow strategy>(SAP driven)

The SAP refined map proposes the generation of an annual cash plan using the *Medium-term Forecasting strategy* and the production of short term cash plan based on Forecasting Short Term Cash Position using Clearing Account, Cash Account and Payment Advice Note strategies. This liquidity management can be seen to be more sophisticated than the current practice modelled in the As-Is map. This suits liquidity management in regions since they are concerned with the availability of liquidity to carry out their plans. Regions are responsible for forecasting and managing their cash in accordance with their budget. Therefore, the refinement of section *<Plan Regional Funds, Manage Regional Liquidity, Regional Cash Flow strategy>* remains identical to the SAP refinement of the corresponding section as shown in Figure 7.

Step 10: Refining the section <Plan National Funds, Manage National Liquidity, National Cash Flow strategy> (SAP driven)
The national level is not involved in the day-to-day liquidity operations and has to do very little liquidity forecasting and management. At the national level the interest is that the financial investments made contribute optimally to the management of liquidity. Thus, it seems

that the SAP refined map is inappropriate because there is no need to have a short-term cash plan which means that *Forecast Short-term Cash Position* is not required for this level. Therefore, only the section *<Start, Generate Cash Plan, Medium-term Forecasting strategy>* of the SAP refined map is needed. Consequently, the *<Plan National Funds, Manage National Liquidity, National Cash Flow strategy>* is equivalent to *<Start, Generate Cash Plan, Medium-term Forecasting strategy>*.

Step 11: Completing the liquidity decentralisation requirement (As-Is & To-Be driven)

The national level is responsible for ensuring financial discipline in the Unit. Therefore, it must monitor that the cash flow in all the regions is in accordance with the national budget. This implies that the regional and national levels agree to the regional cash plans and their consolidation into the national cash plan. This calls for the introduction in the *Matched Map* of the section *<Manage Regional Liquidity, Manage National Liquidity, Consolidation strategy>*(Figure 10).

Step 12: Revising the budget (SAP driven)

The SAP map includes a feed back flow from *Manage Liquidity* to *Plan Funds* using the *Liquidity-based Revision strategy*. This section is relevant for the *Matched Map*. Since transactions occur in the regions, the *Liquidity-based Revision strategy* is introduced to allow flow from *Manage Regional Liquidity* to *Plan Regional Funds*. It should be noted that in the event of revision of any of the four regional budgets, the *Consolidation strategy* will be used to propagate the revision to the national budget. Thus, the section *<Plan Regional Funds, Plan National Funds, Consolidation strategy>* will be triggered.

Further, it can be seen that the two flows based on the *Consolidation strategy* in the *Matched Map*, namely, *<Plan Regional Funds, Plan National Funds, Consolidation strategy>* and *<Manage Regional Liquidity, Manage National Liquidity, Consolidation strategy>* help in obtaining agreement on the formulation and subsequent revisions of the regional and national budgets and regional and national cash plans. Due to this, there is no need for a feed back flow from *Manage National Liquidity* to *Plan National Funds*.

Step 13: Terminating the construction of the Matched Map (SAP driven)

The SAP map is terminated by the section *<Manage Liquidity, Stop, Completion strategy>*. Since the national level is responsible for financial discipline, completion in the Matched Map must flow from *Manage National*

Liquidity. This justifies the introduction of the section <*Manage National Liquidity, Stop, Completion strategy*> in the *Matched Map* (Figure 10).

Step 14: Verification of the Matched Map against the To-Be map (To-Be driven)

The *To-Be map* of Figure 9 makes the distinction between budget management and cash management. This is obviously taken into account in the *Matched Map* by the separation between planning funds and managing liquidity respectively.

The *Delegation to Regions strategy* of the To-Be map has a major effect on the adaptation of the SAP map to the case. This results in a symmetric structure of the flow from planning funds to managing liquidity at the regional and national levels. The co-operation necessary to make the decentralisation work is modelled by the pair of consolidation strategies between these two symmetric strands. The SAP driven construction helps in understanding management of liquidity in the decentralised environment even though decentralised cash plan management is not explicitly recognised in the To-Be map.

The *Year-based Forecasting strategy* of the *To-Be map* is represented by the *Regional Planning strategy* and the *National Financial Investment strategy*. The *Medium-term Forecasting strategy* is represented by *the Regional Cash Flow strategy* and the *National Cash Flow strategy*. The *Quarterly Revision strategy* is implemented as the *Liquidity-based Revision strategy* flowing between *Manage Regional Liquidity* and *Plan Regional Funds*. The *Transaction Recording strategy* is included in the refined map (see Figure 7) of the section <*Plan Regional Funds, Manage Regional Liquidity, Regional Cash Flow strategy*>. It is implemented as *the Clearing Account, Cash Account, and Payment Advice Note strategies*. Finally, the *Completion strategy* remains in the *Matched Map*.

6. Conclusion

We presented an approach which supports the alignment of ERP system functionality to customer requirements. It consists of a process model to show the way ERP requirements match customer requirements. Specifically, this model contains an abstraction step to abstract from ERP system functionality to ERP system requirements thereby (a) establishing a post-traceability link between ERP requirements and functionality and (b) providing an intentional view of what the ERP system can do. The latter is essential to support strategic decisions about the parameterisation of the ERP system to fit the organisation business policy. The former provides a straightforward

way to select the ERP functionality together with its variants conforming to the matched requirements.

The requirements matching process is principally ERP driven. This is the way by which we can minimise the number of deviations from the ERP functionality in a given installation and also benefit from the full power of the ERP functionality. This ERP drive seems to be an essential characteristic of requirement engineering processes of off-the-shelf systems.

Another major difference between traditional RE and that for off-the-shelf systems seems to be that a complete requirements specification is not needed for the latter. Instead, initial incomplete requirements can be progressively refined and detailed to restrict the solution space and converge towards the needed functionality.

The next step in our work is the application of the approach to a large scale industrial project. Also, we expect to lay down guidelines for performing the abstraction of functionality into requirements

7. References
[1] Alvaro-Diaz, Propositions d'outils méthodologiques pour l'étude différentielle d'un projet d'intégration d'un ERP: Application au cas de l'intégration de PeopleSoft au sein du Ministère de l'Emploi et de la Solidarité, Mémoire de DESS 'Systèmes d'Information', IAE de Paris, 2000.

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

[3] S. Assar, S. Si Said, C. Ben Achour, *Un modèle pour la Spécification des Processus d'Analyse des Systèmes d'Information*, Proc. of the INFORSID'2000 Conf., Lyon, France, 2000.

[4] C. Ben Achour, C. Ncube, *Engineering the PORE Method for COTS Selection and Implementation With the Map Process Model*, Proc. Of Workshop on Requirements Engineering For System Quality (REFSQ'2000), 2000.

[5] M. Feblowitz., S. Greenspan, H. Reubenstien, R. Walford, *ACME/PRIME : Requirements acquisition for process-driven systems*, Proc. of the 8th International Workshop on Software Specification and Design, IEEE Computer Society Press, Washington, DC, pp 36-45, 1999.

[6] A. Finkelstein., G. Spanoudakis, M. Ryan, *Software package requirements and procurement*, Proc. of the 8th International Workshop on Software Specification and Design, IEEE Computer Society Press, Washington, DC, pp 141-145, 1996.

[7] G. Fox, S. Marcon, K. Lantner, *A Software Development Process for COTS-based Information System Infrastructure*, Proc. of the Fifth Intl. Symp. On Assessment of Software Tools and Technology, 1997.

[8] M. Jackson, *Software Requirements and Specifications - A Lexicon of Practice, Principles and Prejudices*. Addison Wesley Press, 1995.

[9] Laugée Damien, L'implantation d'un progiciel de gestion intégré dans une entreprise : du paramétrage aux spécifiques utilisateurs. Chez Thomson-CSF AIRSYS

[10] S. Lauesen, M. Mathiassen, *Use Cases in a COTS Tender,* REFSQ'99, Int. Workshop on Requirements For Systems Quality, Hiedelberg, 1999.

[11] N. Maiden, C. Ncube, *Acquiring COTS Software Selection Requirements*, IEEE Software, 15(2), 46-56, 1998.

[12] C. Ncube, N. Maiden., *Guiding parallel requirements acquisition and COTS software selection*, Intl. IEEE Conference on Requirements Engineering, Limerick, Ireland, 1999.

[13] C. Potts, *Inventing Requirements and Imagined Customers: Requirements Engineering for Off-The-Shelf Software*, Proc. Second Intl. Symp. On Requirements Engineering, IEEE Computer Society, 1995.

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

[15] Tran & Liu, *A Procurement-centric Model for Engineering Component-based Software Systems*, Proc. of the Fifth Intl. Symp. On Assessment of Software Tools and Technology, pp 70-80, 1997.