Migration of Legacy Information System based on Business Process Theory

Atsa Etoundi Roger
Departement of Compuer Science
Faculty of Sciences
University of Yaounde I

Abessolo Alo'o Ghislain
Departement of Compuer Science
Faculty of Sciences
University of Yaounde I

Simo Bonaventure Joel
Departement of Compuer Science
Faculty of Sciences
University of Yaounde I

ABSTRACT
The permanent evolution of needs and ICTs on the one hand and taking into account customer satisfaction on the other hand has lead companies to adapt their information systems. Taking these developments into account inevitably leads to the obsolescence of existing systems. In most cases, these legacy systems are not fully documented. This state of affairs renders the migration process very complex. The situation is much more complex in developing countries where only information systems are not documented but, also suffer from a lack of qualified staff capable of maintaining information systems. Till date, the work done in legacy information systems migration is based on documentation of these systems, but do not specify how the documentation was created. In this work, we propose an approach based on the theory of business process and workflows which systemizes reverse engineering of legacy information systems to provide the documentation necessary for migration

General Terms
Legacy Information System Migration

Keywords
Legacy Information Systems, Migration, Understanding, Documentation, Enterprise Modeling, Business Process Theory, Workflows Theory.

1. INTRODUCTION
Technological evolution and changing needs have a significant influence on how organizations must provide services to satisfy customers and remain competitive. Competitiveness and customer satisfaction depends largely on the potential of their information systems. The consideration of these technological evolution and changing requirements [13,14] in the production of goods and services renders information systems obsolete. This situation induces organizations to migrate their information systems to enable the integration of new requirements and taking into account the new technological possibilities.

The consideration new requirements and new technological advances in an information system remains open field of research. Generally, the migration process has two main activities: the conversion of the database and processing of source code for use in a new database and a new development platform [1]. Several approaches have already been set for the migration of legacy information systems, but a few of them emerge. We note in this register Big Bang method [2,10] which is one of the first methods proposed in the area. It recommends the redevelopment of the system from scratch while incorporating new requirements and taking into account of the latest technological opportunities. The methods Database First Approach and Database Last Approach opt for the migration of the database respectively before and after the migration of source code. Both approaches are based on the use of a gateway which enables communication between the legacy system and the new database during migration, the new system and legacy database respectively. Moreover, some methods like Composite Database Approach [2,8] Chicken-Little Strategy [8,11] advocate the duplication of the database and the use of a multi Gateway function as the system obsolete or decomposable, semi-decomposable or not decomposable. This Multi Function Gateway intercepts requests from the legacy system and routes them to the database. If it is updated request, the two databases are synchronized; otherwise the reading may be in either of the two databases. The method Butterfly of J. Bisbal and *al* [3] eliminates the concept of Gateway and is based on the principle that the data and his schema are the most important parts of the legacy information system. After the schema of the new database designed and implemented, the usage of a Data-Transformer enables data migration from the old to the new database. Subsequently six main steps are proposed to guide the migration process, each containing activities that can be performed in parallel.

A survey conducted on the migration of information systems by J. Bisbal and al [2] shows that several aspects of migration such as the choice of technology and architecture of the new system, the validation functionality, management of a migration project, understanding the legacy system information, are still unexplored as they contribute to the facilitation of the migration process. However all the approaches mentioned above and aspects of migration mentioned in [2] based on the understanding of the system, which in turn depends directly on the documentation of the system [15]. In obsolete information systems where documentation is sometimes lacking, the major challenge of the migration of these types of systems is based on the reconstitution of the documentation.

Regarding the specific case of information systems in developing countries, documentation related problem remains preoccupying. In this respect, the migration methods mentioned above are difficult to apply to these information systems. It appears clearly an issue of adaptability of these methods to the proposed migration of information systems in the South. The lack of documentation is in mostly related to lack of qualified human resources [7] needed to monitor the implementation of information systems by qualified organizations. This lack of skilled resources leads software development corporations not to provide all documentation defined in the implementation phase, thus limits understanding of the system. However, according to Ganti and Brayman [5], the understanding of the migration is subject to the starting point of migration. Generally these software development corporations take control of the system and reduced the proprietary organization to a simple user who has no technical knowledge or documentation on the setting up and designing of their information system. The approach proposed in this work provides for the reconstitution of legacy information system documentation required for its migration. This approach is based on the theory of modeling business processes and workflows defined in [16].

The rest of this paper is organized as follows: Section 2 provides a summary of the theory of business processes modelling and workflows modelling. Section 3 presents the migration approach based on this theory. Section 4 presents the application of this approach in the reconstruction of the technical documentation of a database, and Section 5 concludes the work and highlighting some perspectives as future works.

2. BUSINESS PROCESS AND WORKFLOW THEORY

In this section, we present in an incremental manner concepts that are borrowed for the theory proposed by Atsa and *al* [16,19, 20,21] for the modeling of business process and workflows within an organization in order to manage the satisfaction of different stakeholders. We then move from basic to complex concepts. Those concepts are suitable to tackle the legacy information system reengineering.

2.1 The Environment Description Model

The environment is considered as a set of different metrics whose value may change [17]. These metrics are primitive Boolean observers denoted by Observer. The associated value of each observer depends on the current state of the environment.

Formally, an environment *E* is defined as a couple $\langle \Theta, S, val \rangle$ where:

- Θ is a non empty set of observers;
- *S* is a non empty set of sates;
- $val : \Theta \to (S \to Bool)$ is a function which describes the behaviour of observers.

In this rest of summary, $val(o)(s)$ is denote by $s(o)$ where *s* denotes a state and *o* an observer, $s(o)$ is the value of the observer *o* in the state *s*. Given a state *s*, the set of observers whose value is true defines the characteristic of *s* and is represented by $s_C = \{o \in \Theta, s(o) = true\}$.

Given two states *s1* and *s2* of the set of states *S* of the environment *E*, the set of observers whose associated values are not the same is defined from the characteristics of the two states. This set is called gap between *s1* and *s2* and is denoted by $s1 \bullet s2 = (s1_C - s2_C) \cup (s2_C - s1_C)$.

Given an environment *E*, the observers in Θ define the alphabet that permits to reason about events that occur on *E*. The language defined from this alphabet is denoted by the set of conditions or formulae *C*. A condition $c \in C$ is an assertion over observers and is defined as a first order formula. The basic elements of *C* are therefore all the observers of Θ. The elements of *C* are formed by the following:

$$\begin{cases} if\ \ o \in \Theta, then\ o \in C \\ if\ \ o \in \Theta, then\ \neg o \in C \\ \text{if}\ \ o_1, o_2 \in \Theta, then\ o_1 \vee o_2,\ o_1 \wedge o_2, o_1 \Rightarrow o_2 \in C \end{cases}$$

A condition *c* can be decomposed into a set of observers $+c$ whose values are evaluated to true and a set of observers $-c$ that are evaluated to false. The two sets do not have any common element i.e. $+c \cap -c = \varnothing$

Given a condition $c \in C$ and a state *s*, *c* is satisfied within the state *s* if the result of its evaluation is true, i.e. $s(c) = true.$.

2.1.1 State of an Environment

A state is a snapshot of an environment within a time [17]. From this snapshot facts are observed. Some of these facts or features of a state are true or false at this particular time. These facts are represented as some equivalent of predicate calculus formulae. We shall refer, somewhat loosely, to these facts and relations as attributes of a state. In a rigorous manner, let *F* be a set of formulae, and s be a state, then s is a subset of *F* i.e. $s \in F$ **.**

In general, let *S* be a set of states, according to the definition of a state, $(S, \subseteq)$ is a partial ordered set. The model are not dealing with any kind of set of states, it is interested with *S* having a least state, $\perp_S$, known as initial state of a business process or workflow from which the execution can be started. This initial state is therefore contained in all states of *S* i.e for all $s \in S$, $\perp_S \subseteq s$. In the meantime, *S* is required to have a least upper bound $\bigcup_S$ known as a state where the goal of the business process is satisfied.

2.2 Knowledge Model

In [21], a goal oriented approach- for the definition of a business process requirement model, integrating their level of importance and constraints inherent to these requirements is presented. The level of importance of a goal is the credit which the user associates to this goal. Constraints are non-functional requirements related to what this goal must satisfy. The approach that was proposed revolves around four main activities: requirement elicitation, selection of different goals, transformation of requirements into knowledge bits and finally the development of the requirement model. However, the proposed knowledge model needs to be reviewed according to the concerns outlined in the previous section. That is why the new model will put a strong link between the importance associated to knowledge and experience level of its author that is who emits or has knowledge. Moreover, according to [21], it is the author of knowledge who defines its degree of relevance while this degree of relevance should be deduced from other knowledge components. For this, [21] define knowledge as a tuple $<k, Ag, Ex, y, w, l, d, v>$ where: *k* is the name of the knowledge; *Ag* is the name of the agent who expressed the knowledge; *Ex* is the experience level of *Ag*; *y* the context in which the goal is defined; *w* is the goal; *l* the business rule; *d* execution constraints; *v* the level of importance of the goal.

2.3 QoS Model

The quality of service denoted by QoS represents the performances of the service which determine the level of satisfaction projected for the recipients of the services [16]. The level of satisfaction is defined as a set of properties, criteria, characteristics and performances of the services delivered to the customers. Several works are made in this field, each one defining a specific set of criteria specified in order to measure the QoS. In the literature, there is no consensus yet on the definition of the set of common criteria to evaluate the quality of service delivered in the organizations [16, 19]. The evaluation criteria are defined according to the objectives and specificities of each company. The concept also defines an abstract model which gives the semantics of the quality of service.

Definition 2.1
Let Cr be a set of criteria considered in the evaluation of the quality of service, Val the set of values that can be assigned to these criteria, and f a map defined by $f: C \rightarrow Val$, the QoS is defined by (C, Val, f).

Given two QoS *q1* and *q2* such that $q1 = (Cr1, Val1, f1)$ and $q2 = (Cr2, Val2, f2), q1$ and *q2* are compatible and denote by $q1 \Delta q2$ if and only if $C1 = C2$ and $Val1 = Val2$. When *q1* and *q2* are compatible, *q1* is better to *q2* and denote $q1 \subseteq q2$ if and only if $\forall c \in C1, f1(c) \leq f2(c)$. $(\Phi, \subseteq)$ is use to denote the partial ordered set of compatible qualities of services [16].

2.4 Task Description Model

A task is an atomic activity that cannot be split into smaller activities [16, 17, 20]. The performance or execution of a task transforms the state of the environment into another state. A task is therefore an action within a state of an environment. Before a task can be executed, the state of the environment should satisfy a specific condition called pre condition, and when this execution is completed another condition, called post condition is satisfied. For a task to be executed within an organization which will be defined later, the knowledge required for its performance is captured. This knowledge depends on the context within which the execution can take place. For each of the associated contexts is defined a set of knowledge bits and quality of service to obtain after the execution of a task. A task is formally defined in [19] by a tuple $<nt, PP, f_m, g_m, Cx, KBx, Qx>$ where *nt* denotes the name of the task, $PP = Pre \times Post$ where Pre denotes the set non empty set of preconditions within which its execution can be carried out, and $Post$ the set of post conditions that are obtained after the execution, Cx a non empty set of contexts within which the task can be executed, KBx a non empty set of knowledge bits used for the better understanding and performance of the task, Qx is a quality of service to be reach after the execution of nt. f_m, and g_m are maps defined respectively by:

$$\begin{cases} f_m : Cx \rightarrow PP \\ g_m : Cx \rightarrow KBx \end{cases}$$

If *c* denotes a context of Cx, then *c* is a restriction of the environment Θ, that is $c \subseteq \Theta$. The action of a task within an environment is to transform its current state into a new one. When $<nt, PP, fm, gm, Cx, KBx>$ is a task, *s* a given state where the precondition $pre(PP)$ is satisfied i.e $s(pre(PP)) = true$, the action of *t* in the state *s* is the new state $t(s)$ which satisfies the post condition $post(PP)$ i.e $t(s)(post(PP)) = true$ *e*. In general, the action of a task *t* within the state *s* is characterized by the observers of *s* whose value has been modified.

Definition 2.2 *(Task action)*
Let $E = \langle \Theta, S, val \rangle$ be an environment, *s* a given state and *t* a task whose pre condition is satisfied in *s*, then the action of *t* in *s* denoted by t_s and is specified by $t_s = \{o : \Theta, s(o) \neq t(s)(o)\}$.

A task will be represented when there will be no ambiguity by its name *t* and $pre(t)$ respectively $post(t)$ will denote respectively its pre and post condition. Based on the post condition of a task *t*, and the state *s* where $s(post(t)) = true$, we conjecture that $t_s = +post(t) \cup -post(t)$ [16].

Definition 2.3 *(Conflicting Tasks)*
The action of tasks within an environment can be conflicted as many tasks can modify the same observers at the same time [16]. To this end, t_1 and t_2 are conflicting tasks in the state s, and we denote it by $overlap(t_1, t_2, s)$, if and only if:

$$\begin{cases} s(pre(t_1)) = s(pre(t_2)) = true \\ + post(t_1) \cap - post(t_2) \neq 0 \\ + post(t_2) \cap - post(t_1) \neq 0 \end{cases}$$

Definition 2.4 *(Orthogonal Tasks)*
Let $t_1 = <nt_1, PP_1, fm_1, gm_1, C_{X1}, KBx1, Qx1>$ and $t_2 = <nt_2, PP_2, fm_2, gm_2, C_{X2}, KB_{X2}, Q_{X2}>$ denote two tasks, t_1 and t_2 are said to be orthogonal if and only if t_1 and t_2 require the same knowledge in order to be processed whenever the processing context is differed, i.e $C_{X1} \neq C_{X2}$ *and* $KB_{X1} = KB_{X2}$ [19].

Definition 2.5 *(Shift)*
Let SoT be a none empty set of tasks and *s* a given state, a shift denoted by Shf is a couple $Shf = <s, SoT>$ composed with the state *s* and the set of non conflicting tasks SoT within *s* [16,19].

Formally, let $Shf = <s, SoT>$ be a shift, the following properties are satisfied:

$$\begin{cases} S \circ T \neq \varnothing (1) \\ \forall t \in S \circ T, s(pre(t)) = true (2) \\ \forall t, t' \in S \circ T, t \neq t' \Rightarrow overlap(t, t', s) = false (3) \end{cases}$$

Let $Sht = <s, SoT>$ be a shift, the simultaneous actions of SoT in *s*, denoted by $ts(s)$, is captured by the set of observers whose values are modified within *s*, that is:

$$S \circ T(s) = \bigcup \begin{Bmatrix} o \in \Theta : \\ o \in -post(t_i) \end{Bmatrix}, t_i \in S \circ T$$

Definition 2.6 *(Chain)*
A chain is an execution path of tasks, according to their actions in states and their triggering conditions is denoted by $P = \prod_{i=1}^{n} Sht_i$, and is specified as a finite sequence of shifts where *n* represents the length of the sequence [16,17,19].

Let P be a path of length $n>1$, and $sh_k=\langle s_k,st_k\rangle$, $sh_{k+1}=\langle s_{k+1},st_{k+1}\rangle$ notes respectively the shift in the range k and $k+1$, the state s_{k+1} is the resulting state after the execution of the set of tasks st_k i.e $s_{k+1}=st_k(s_k)$. When there will be no ambiguity, the shift of the range k of the path P will be denoted by $P(k)$.

Let $Sht_k=\langle S_k,S\circ T_k\rangle$ and $Sht_{k+1}=\langle S_{k+1},S\circ T_{k+1}\rangle$ be two shifts where $Sht_k=S\circ T_k(s_k)$, the difference between the states s_k and s_{k+1} is denoted by $\overline{s_k+s_{k+1}}$ and is defined as follows: $\overline{s_k+s_{k+1}}=S\circ T_k(s_k)$.

Lemma 2.1
Let *p* be an execution path and $t\in S\circ T(p(k))$ with $k\le length(p)$ then there will always exist *m* such that $m>k$ and $S(p(m))(post(t))=false$.

Lemma 2.2
Let *p* be an execution path then $S\circ T(p(length(p)))=\varnothing$.

Definition 2.7 *(State ordering)*
Let P be a path of length $n>1$, and $Sht_k=\langle S_k,S\circ T_k\rangle$ and $Sht_{k+1}=\langle S_{k+1},S\circ T_{k+1}\rangle$ be two consecutive shifts in *P* with $k<n$ then $S_k\subseteq S_{k+1}$ specifies the fact that the set of observers modified in S_k after the actions of SoT are contained in the set of observers of S_{k+1} with the same values.

Lemma 2.3
Let P be an execution path, *S* the set of states of P, then $(S,\subseteq)$ is completed partial ordered where the least upper bound state in the last state of P and the least state is the first state of P.

The modeling has to ensure that the execution of a task *t* will stop at a certain time. In order to do so, the set of observers that should be modified by *t* must contain partially or totally in the observers forming its pre condition $(-pre(t)\cup -pre(t))\cap(-post(t)\cup +post(t))\neq\varnothing$.

From the definition of the execution path of tasks, we specify the relation within the set *T* of tasks based on the set *S* of states. This relation is denoted by $\trianglelefteq$.

Definition 2.8 *(Ordering of Tasks)*
Let *T* be a set of tasks, and *t1* and *t2* be two tasks of *T*, we write $t1\trianglelefteq t2$ if and only if for all chain CH such that if n_{t1} and n_{t2} denote respectively the maximum range of *t1* and *t2* in CH, then $n_{t1}\le n_{t2}$. This relation has the following properties:

1. reflexivity: $t\trianglerighteq t$ this simply means that the task *t* belongs to the chain CH;
2. antisymetric: if $t1\trianglerighteq t2$ and $t2\trianglerighteq t1$ in the chain *D* then $t_1=t_2$. By convention, there will always exist a path from each task to itself;
3. transitivity: obviously if in the chain CH, $t1\trianglerighteq t2$ and $t2\trianglerighteq t3$ then $t1\trianglerighteq t3$.

Lemma 2.4
The set of tasks *T* associated with the relation previously defined $\trianglerighteq$, i.e. $(T,\trianglerighteq)$, forms a complete partial ordered set.

2.4.1 Palette

Let *E* be an environment, and *S* be a set of different states that *E* may reach according to the actions of tasks *T*, then a palette *P* is a couple $<E,S\to S>$ [16]. The set of functions $S\to S$ will be denoted by *T*, the set of tasks of the palette. $P(E)$ and $P(E)$ will denote when there will be no ambiguity, the environment and the set of tasks of the palette *P* respectively.

The actions of the set of tasks *T* of the palette *P* in the environment *E* are to change at least once the value of each observer of Θ in *E*. To this end, the consecutive actions of a non empty set of tasks within an environment may not modify all the observers in this environment. The set of observers whose value are not changed during the execution of any given none empty set of tasks will be abstracted from all the possible states of the environment, i.e.

$$\forall o\in\Theta\begin{cases}\exists t_1\in T, o\in -post(t_1)\\ \exists t_2\in T, o\in +post(t_2)\end{cases} or$$

Given a palette *P*, according to the environment changes within organizations and the different executions of tasks that can take place, different ways in which tasks can be executed have to be captured. $SP_{\mathbf{P}}$ is use to specify the set of execution paths that can be obtained from a palette *P*.

Lemma 2.5
Let *P* be a palette, $s\in S(P)$ a given state of the environment $E(P)$ of *P*, there will always exist a path $p\in SP_P$ such that $s\in S(p)$, where $S(p)$ denotes the set of states of the path ***p*.**

Lemma 2.6
Let $P=<E,T>$ be a palette, and $t\in T$, there will exist an execution path $ch\in SP_P$ where SP_P denotes the set of possible execution paths of *T*, $ch(n)=\langle s_n,S\circ T_n\rangle$ such that $t\in S\circ T_n$.

2.5 Business Process Model

A business process is a collection of activities or tasks designed to produce a specific output for customers [16,17,19]. It implies a strong emphasis on how work is done within an organization in order to deliver a particular service. A process is thus a specific order of work activities across time and space, with a beginning, an end, and clearly defined inputs and outputs. The output is the reason the organization does this work and is defined in terms of the benefits this process has for the organization as a whole.

Definition 2.9 *(A service)*
A service is the characteristic of a business process and is defined as a composition of a set of criteria that characterize what is delivered within an organization, where each criterion is represented by an observer [16,17,19].

The model of a business process is defined as a couple $<P,G>$ where *P* is a palette and *G* the service to be achieved. According to the definition of the palette, the ordering of tasks is captured explicitly by their pre conditions and the states of the environment within which their execution is being carried out.

This approach reduces the number of patterns to be used in order to capture various ways tasks can be ordered. That this is the main difference between this modeling approach and other BPM theory papers presented in the literature. In these works, the Workflow Management Coalition [18] has identified four basic control structures for workflows: *OR-SPLIT*, *OR-Join*, *AND-Split*, and *AND-Join*. More control structures have been identified by Van der Aalst in [6].

Lemma 2.7
There will always exist a state s_{lub} such that when it is reached, other states cannot be reached. This state is called a least upper bound state of the associated business process.

Lemma 2.8
There will always exist a state s_{ini} from which the execution of the business process starts. This state is called a least state of the associated business process.

For each service associated to a given business process, a set of qualities of service is defined to deal with the daily work and the competitive pressure of the network economy.

Definition 2.10 *(Well Defined Business Process)*
Let, $BP{=}{<}P,G{>}$ be a business process, *BP* is well defined if and only if all the observers that form its goal (service) are contained in the set of observers of the environment *E* i.e. $-G\cup+G\subseteq\Theta(E)$

Definition 2.11 *(Well Formed Business Process)*
Let $BP{=}{<}P,G{>}$ be a business process, BP is said to be well formed if and only if each execution chain SCH reaches the least upper bound state s_{lub} which satisfies the service *G* i.e.

$$\begin{cases}\forall ch\in SCH, n_{ch}\in N, s_{\text{lub}}\in S\\ n_{ch}=length(ch)\\ s_{\text{lub}}\\ s_{\text{lub}}(G)=true\end{cases}$$

More formally, let SCH be the non empty set of different chains that can be obtained from a business process BP, and $CH{\in}SCH$ with the length n_{CH} such that the n^{th}_{CH} state s_{lub} of CH satisfies *G* i.e. $s_{lub}{=}true$.

Definition 2.12 *(Deadlock- and Livelock-Free)*
Let BP be a business process, BP is deadlock- and livelock-free if and only if it guarantees that every execution chain reaches its least upper bound state satisfying the goal of the business process BP .

Theorem 2.1
Let BP denote a business process such that BP is well defined and well formed, then BP is deadlock-free and livelock-free.

Proof: By the definitions of well formedness and well definedness of a business process which states that the least upper bound of the state of a business process is reached and that this least upper bound state satisfied the goal of the business, the described business process model is deadlock and live lock free.

All the execution paths of a business process start from the same state denoted by s_{ini} . It can be easily being shown that the set of states S_{BP} associated with the ordering relation $\subseteq$ as defined previously is completed partially ordered.

2.6 Human Actor Model

There are many types of agents participating in the processing of tasks within an enterprise for the achievement of customers' needs. The enterprise system dealing with the processing of tasks is a hybrid system including hardware components with embedded software, the human actors interacting with the hardware and the organization. An organization is an arrangement of human actors purposefully organized to carry out a certain mission, which, in its turn, adds a dimension to the quality of service [16]. The hardware components have been designed to play specific roles and functions in the process chain, and can hardly be moved among different roles in the enterprise as it is done for human actors. The modeling is not dealing with hardware but with human actors who can significantly influence the quality of service according to their skills and associated experiences. We model the skill of a human actor by (Sk,Tks,mch) where Sk is the set of competences, Tks the set of tasks and *mch* a map that gives for each competence $cp\in Sk$ the set of tasks $mch(cp){\in}Tks$ that can be processed based on *cp* with $mch(cp)\neq\varnothing$ The structure (Sk,Tks,mch) will be represented by Sk when there will be no ambiguity. Based on the organization put in place, the set of tasks assigned to a human actor are kept in a diary.

A diary is described by the set of tasks and the set of time intervals within which there are processed [16]. It is important that the set of time intervals in the agenda be defined such that it does not allow the overlapping of time intervals.

Let $Pds{=}(TI,\subseteq,\cap,\Delta)$ be a set of time intervals such that $(TI,\subseteq)$ is a partial ordered set with ∂ the smallest time interval, $\cap$ and Δ be two maps defined as follows $\cap: TI\times TI\rightarrow TI$ and $\Delta: TI\times TI\rightarrow Boolean$, *t1* and *t2* be two time intervals of *TI*, *p1* and *p2* overlapped if and only if there exists a time interval *p3* such that:

$$p1\cap p2=t3\Rightarrow\begin{cases}p3\Delta p1\wedge p3\Delta p2\\ p3\subseteq p1\wedge p3\subseteq p2\end{cases}$$

where $\cap$ and Δ define respectively the intersection and the overlapping relationship. The set of time intervals is represented when there is no ambiguity, by Pds . Based on the concepts of tasks and time interval, the diary concept is modeled by ${<}Tks,Pds,g{>}$ where Tks is the set of tasks, Pds the set of associated time intervals, and *g* a map defined by $g: Tks\rightarrow Pds$ such that $\forall t1,\, t2{\in}Tks,\, t1{\neq}\, t2{\Rightarrow}\neg(g(t1)\Delta g(t2))$.

Definition 2.12 *(Human Actor)*
A human actor is defined in [19] by ${<}Sk,Ex,f,Dy,Id{>}$ where Sk is its set of skills, Ex the set of associated experiences, *Id* its identification, Dy its associated diary, and *f* a map which defines for each skill $sk{\in}Sk$ its associated experience $f(sk){\in}Ex$.

2.7 Workflow modeling

A workflow is defined by (Ts, Es,Ps,h, f_{em}, Q)+ where Ts is the set of none conflicting tasks, Es the set of employees dealing with the processing of Ts within the time intervals Ps to obtain the quality of service Q , *h* is the map $Ts\rightarrow Ps$ which defines for each task *t*, its time interval $h(t)$ within which it is processed, and *f* a map that gives for each task *t* the employee $f_{em}(t)$ who is charge of its processing [19]. The two maps *h* and *f* are required to be two isomorphism as each task is required to be associated to a time interval within which its execution will take place, and should also be assigned to a specific employee for this performance. The quality of service Q is such that:

$Q=\sum_{i=1}^{n} q_i$ where q_i is the quality of service obtain after the execution of task $t_i \in Ts$ and *n* the number of task in Ts .

Based on the fact that the satisfaction of customers is one of the challenges that enterprises are required to guarantee, in the modeling of the workflow, we require that employees who are involved in the processing of tasks have the necessary knowledge to carry out these tasks. Therefore, if *t* is a task to be carried out by the employee $f_{em}(t)$, and $kb_{em}(t, f_{em}(t))$ his knowledge associated for the processing of *t,* there will exist at least a context *c* within which *t* can be processed such that the knowledge $bk(t,c)$ required for its processing verifies the following constraint $bk(t,c) \subseteq kb_{em}(t, f_{em}(t))$.

2.8 Enterprise Model

An enterprise is a structure dealing with the service delivery of customers based on a certain quality of service. This structure is organized in terms of business processes that are carried out, employees in charge of the processing of the associated tasks, and the resulting workflows.

Definition 2.13 *(WorkStation)*
A workstation *wk* is a position within an enterprise defined by $(Tks, KBs, \omega_{tk}, \omega_{pk})$ where Tks is the set of tasks to be carried out by a human actor appointed at this position, KBs is the set of knowledge bits required for the performance of tasks Tks , and ω_{tk} is a map which gives for each task tk , the tacit knowledge $\omega_{tk}(tk)$ built by the former employees in this position, $\omega_{pk}(tk)$ defined the critical knowledge required for the processing of tk based on the execution context [19].

The model in [19] does not accept workstation $wk=(Tks, KBs, \omega_{tk}, \omega_{pk})$ with an empty set of knowledge bits related to its given task, i.e $\forall tk \in Tks,\ \omega(tk) \neq \varnothing$. Moreover, if wk denotes a workstation, $tasks(wk)$ denotes its associated set of tasks.

Definition 2.14 *(Enterprise)*
An enterprise Org is modeled by $(Io, BPs, Emps, WFs, WKs, f_{ewk})$ where Io is its identification, BPs is the set of its business processes that can be run, $Emps$ its set of employees who participated in the processing of tasks defined in various business processes, WFs its set of workflows defined for the achievement of customer's needs, WFs the associated workstations, and f_{ewk} denotes a map which gives for each employee $ag \in Emps$, the position $f_{ewk}(ag)$ that he is appointed [19].

Based on the human actors working in a given enterprise and their availability and the services required by customers, employees involved in different workflows associated to a business process will not necessary be the same. To this end, according to their skills, the quality of service delivered may be different. The criteria for the evaluation of the quality of service will then some time be associated with minimum values when tasks will be processed by staff with minimum experience. More-over these values will be maximal when staff with maximum experience has been involved in the processing of tasks. The set of quality of service associated to a given business process will therefore have two specific qualities of service Q_{min} and Q_{max} which have the following properties.

Lemma 2.9 Let $Q_{min}=(C, Val, f_{qmin})$, and $Q_{max}=(C, Val, f_{qmax})$, be minimal and the maximal quality of service of a business process (P, Φ) then $\forall p=(C,V,fp) \in \Phi,\ c \in C,\ f_{qmin}(c) \le fp(c)$, and $\forall q=(C,V,fq) \in \Phi,\ c \in C\ ,\ fq(c) \le f_{qmax}(c)$.

2.9 Task processing

Based on the assignment of tasks done by the resource manager within an organization for the achievement a given customer goal, employees process these assigned tasks based on their own experience and the knowledge associated to these tasks. According to the context within which the performance of the tasks is taking place, the processing can be done straightforward if the knowledge related to the task is adequate for its processing within this context. The processing sometime will not be done straightforward as the knowledge related to the performance of the task is not enough. When it is the case, the employee will use his tacit knowledge, on the one received from more experimented employees, in order to process the task. In order to keep track of this new way of carrying out this task, the defined information should be stored for further use. For this end, the knowledge of the so called task should be updated. In order to take this into consideration, the modeling of workflow must take into account the processing of tasks by employees. Let tk be a task that is processed by an employee using the knowledge kb in the context cx , the task tk change the state after its performance based on the fact that, the knowledge associated to this context is updated by the knowledge used for its processing i.e $gm(cx,tk)=gm(cx,tk') \cup kb$ where $gm(cx,tk)$ denotes the set of knowledge required for the processing of the task tk [19].

3. DOCUMENTATION RECONSTITUTION BASED ON THE MODELING OF BUSINESS PROCESS AND WORKFLOWS

An information system is often seen as an interaction between man, algorithms implemented, data and technology. In other words it is likened to a computer program. However, in broad sense, this term refers to a combination of applications, procedures, resources, technologies and personnel organized to facilitate the monitoring, decision making, planning and coordinating the production of goods and services within an organization [9]. This vision of the information system shows that it occupies a very important role within an organization. The business model given in [19] is built on the basis of business process models, workflow, knowledge bits and human actors. To use this model in the migration process, we define application model and information system model based on [16,17,19]. These models shall be used to reconstitute the documentation of the information system.

3.1 Information System Description Model

Definition 3.1 *(Application)*
An application is a set of business process from which are defined workflows and a set of documentation.

Therefore, the conceptualization of an application $APPL$ is given by $APPL = \langle BPs, WFs, Dc, WFs^{BPs}, Dc^{BPs} \rangle$ where:

$$\begin{cases} BPs \text{ is the business process supported by } APPL; \\ WFs \text{ represents all the workflows associated with } BPs; \\ Dc \text{ is the set of document off } APPL; \\ WFs^{BPs} \text{ is a set of functions } f \text{ which for any business process} \\ \quad bp \text{ is associated with to the set } f(bp) \text{ of the different} \\ \quad \text{workflows that are defined from } bp; \\ DC^{BPs} \text{ the set of functions } g \text{ that for each business process} \\ \quad bp \text{ is associated the documentation } g(bp) \text{ with} \\ \quad \text{the constraint } g(bp) \in Dc. \end{cases}$$

In addition $g(bp) = \bigcup_{i=1}^{m} KBx_i$ with m the number of tasks associated with bp, and KBx_i the knowledge bits linked to the t_i of bp. thus, Dc is formally defined by $Dc = \bigcup_{k=1}^{p} g(bp_k)$ where k representing the number of business processes associated to $APPL$.

Definition 3.2 *(Information System)*
An information system is a union of applications of different business process in an organization. An information system denoted IS is formalized as follows:

$IS = \bigcup_{i=1}^{z} APPL_i$ with z being the number of applications of IS.

Applying this modeling, the documentation of a legacy application can be restored by making a description of the different workflows supported by the application. Incrementally, we can reconstruct the documentation of an information system. From this documentation, the existing approaches of migration can be applied to the migration of the information system in question.

4. CASE STUDY: LEGACY DATABASE DOCUMENTATION RECONSTITUTION

A database is a collection of structured information about a subject or a particular purpose. Its construction consists of different phases that begin with the determination of its purpose and its use. All information required of the database is listed here; we determine from there the real world entities for which you want to store in the data and the data associated with each entity. The determination of the different entities, groups information in order not to duplicate them in the database. It is recommended to keep the information on each entity so that they are kept in isolation from others. The determination of the necessary attributes enables the choice of information relative to de concern entity and the domain of this information. The last step is the determination of relationships between entities which allows the grouping of related information in a meaningful and structured manner.

A database thus appears as a collection of data represented by a set of attributes of an application domain, where attributes are in a domain and have constraints, the semantic relationships between these attributes give structure to data and the consistency of the structure is ensured by the constraints between attributes.

Based on the theory of business process and workflow defined in [16,19,20,21], the various elements constituting a database is derived as follows:

Definition 4.1 *(A Database Attribute)*
An attribute is considered to be an observer of the environment. The set ATT of attributes of a database is such that $ATT = \bigcup Ob$ where Ob is an observer of the environment.

Definition 4.2 *(Constraints on an attribute)*
Constraints associated with each attribute refer to different constraints of the associated observer. Constraints related to the set of observers of the environment of the task constitute the pre and post condition of the task. The pre and post conditions are such that $pre = post = Cond^{OB}$ is the set f of function, which for an observer ob combines the associated condition $f(ob)$. The set of constraints Cr_a on an observer is given by $Cr_a = \bigcup_{k=1}^{p} f(ob_k)$ p is the number of pre and post condition in which the observer participates. The set of constraints on the attributes is then formalized by $Cr = \bigcup_{j=1}^{z} Cr_{aj}$ with z being the number of attributes and Cr_{aj} all the constraints of the attribute j.

Definition 4.3 *(Constraints between Attributes)*
They represent the structure of the database and are given by the schema of the database defined using the DBMS, used in the implementation of the legacy database. All of these constraints are given by $U_{ATT} = \bigcup_{k=1}^{m} \left(\prod_{i=1}^{n \in \mathbb{N}^*} A_{ki} \right)$ with $A_{ki} \in ATT$, m the number of relations and n the arity of the k^{th} relation.

4.1 Technical Documentation of a Database

A database manipulated by an application of a legacy information system is modeled by: $DB = \langle ATT, Dom, Cr, Dom^{ATT}, Cr^{ATT}, U_{ATT} \rangle$ where:

$$\begin{cases} DB \text{ is the name of database;} \\ ATT \text{ is the set of attributes;} \\ Dom \text{ is the set of field's attributes and is deduced from the} \\ \quad \text{different processing of tasks of all workflows;} \\ Cr \text{ is the set of contraints on } ATT; \\ Dom^{ATT} \text{ is the set of function } h \text{ that for each attribute } att \\ \quad \text{associates the field } h(att); \\ Cr^{ATT} \text{ is the set of function } k \text{ that for each attribute } att \\ \quad \text{that associates a set of constraints } k(att); \\ U_{ATT} \text{ represents the set of constraints between the} \\ \quad \text{various elements of } ATT. \end{cases}$$

Modeling the database as such enables the understanding of the organization and data semantics. We can now apply a data migration approach like that proposed in [22] which, like all others, requires prior understanding data.

5. CONCLUSION

The understanding of an information system is one of the most important steps in a migration process. Misunderstanding is caused by the lack of necessary documentation on the migration of information systems when they become obsolete. This lack of documentation is source of numerous problems faced by these systems during maintenance operations and interoperability with other systems. Maintaining these types of systems within an organization is not consistent with the evolution of different needs and imperatively requires migration to a new environment. The complexity of the migration process of these systems has led researchers to develop a number of approaches to facilitate this. Despite their differences, almost all of these approaches based on existing documents of the information system and concludes with the difficulty to migrate non-documented systems. Till date, very few studies exist which lead to the reconstitution of the documentation of non documented legacy information system.

Based on the theory of business process modeling and workflows modeling within organizations, this paper presented an approach to reconstruct the documentation of an obsolete information system. The method is based on the concept of task from which various data necessary for its processing are specified. This data includes the knowledge bits necessary for its processing, observers which are the objects of its processing environment, pre and post conditions of the task and relationship-induced by modifications of observers and preconditions of the task. The approach defined in this way was applied to the reconstruction of the technical documentation of legacy database with no documentation. This case study, defined a high-level abstract model of a database.

The approach proposed shall be validated by applying it on other case studies. After this validation phase, we shall setup a tool based on this method. These are some perspectives related to this work.

6. REFERENCES

[1] Jean-Luc Hainaut, A. Cleve, J. Henrard and Jean-Marc Hick. 2008. Migration of Legacy Information Systems.

[2] J. Bisbal, D. Lawless, B. Wu, and J. Grimson. 1999. Legacy Information System Migration: A Brief Review of Problems, Solutions and Research Issues. Technical Report.

[3] J. Bisbal, D. Lawless, R. Richardson, D. O'Sullivan, Bing Wu, J. Grimson, V. Wade. 1997. The Butterfly Methodology: A Gateway-free Approach for Migrating Legacy Information Systems.

[4] W. Hasselbring, R. Reussner, H. Jaekel, J. Schlegelmilch, T. Teschke, S. Krieghoff. 2004. The Dublo Architecture Pattern for Smooth Migration of Business Information Systems: An Experience Report.

[5] N. Ganti and W. Brayman. 1995. Transition of Legacy Systems to a Distributed Architecture. John Wiley & Sons Inc.

[6] W. M. P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. 2000. Workflow patterns. Technical Report wp 47, BETA Research Institude.

[7] Richard Heeks. 2002. Information Systems and Developing Countries: Failure, Success, and Local Improvisations. The Information Society.

[8] M. Brodie and M. Stonebraker. 1995. Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach. Morgan Kaufmann, San Francisco.

[9] http://fr.wikipedia.org/wiki/Management_du_système_d'information. 2011

[10] A. Bateman and J. Murphy. 1994. Migration of Legacy Systems. School of Computer Applications, Dublin City University, Working Paper CA-2894.

[11] M. Brodie and M. Stonebraker.1993 .DARWIN: On the Incremental Migration of Legacy Information Systems. TR-022-10-92-165 GTE Labs Inc.

[12] Anderson S., Felici M. 2001. Requirements Evolution from Process to Product Oriented Management. In Proceedings of Profes. 3rd International Conference on Product Focused Software Process Improvement, Kaiserslautern, Germany, September 10-13, 2001, LNCS 2188, Springer-Verlag.

[13] Harker, S., K.Eason, and J. Dobson.1993. The change and evolution of requirements as a challenge to the practice of software engineering. In IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press.

[14] Päivi Ovaska. 2004. Measuring Requirement Evolution. A Case Study in the E-commerce Domain. ICEIS - information systems analysis and specification.

[15] Dumitru Oprea, Gabriela Mesnita. 2006. The Information Systems Documentation – Another Problem for Project Management.

[16] Atsa E. R., Fouda M. and Abessolo G. 2010. A Denotational Semantics Methodology (DSM) Approach for Business Processes Modelling. IJCA.

[17] Atsa E. Roger and Marcel Fouda. 2002. An Abstract Model for Workflows and Business Processes. CARI 2002, pages 239–247.

[18] D. Hollinsworth. 1994. The workflow reference model. Technical Report TC00-1003, Workflow Management Coalition, http://www.aiai.ed.ac.uk/WfMC/.

[19] Atsa E. R., Fouda M. Atouba C. L. and Abessolo G. 2010. Knowledge management driven business process and workflow modeling within an organization for customer satisfaction. IJCA, Vol. 2 (12), 2010, 7350-7362.

[20] Atsa E. R., Fouda M. and Abessolo G. 2011. A Formal Framework for Business Process Modeling. I JCA, Volume 13– No.6.

[21] R. Atsa Etoundi, M. Fouda Ndjodo, Christian Lopez Atouba. 2010. A Goal Oriented Approach or the Definition of Business Process Requirement Model. IJCA.

[22] Bing Wu, J. Bisbal, D. Lawless, R. Richardson, D. O'Sullivan, J. Grimson, V. Wade. 1997. Legacy System Migration : A Legacy Data Migration Engine.