ISSN 2229-5518

ATSERO Method: A Guideline for Business Process and Workflow Modeling within an Enterprise

Roger Atsa Etoundi

Abstract— In the fields of business process and workflows modeling, a wide range of techniques have been defined and used. Despite the popularity of some of them, there is no consensus on the modeling standards and concepts. However, there are many perspectives that need to be taken into consideration for a better management of workflows within an enterprise such as process, organization, information, operation and the quality of service. In this paper, a new approach, so called *ATSERO Method*, is proposed. The method is based on Formal Method, Domain and Requirement Engineering, is presented. This method describes several salient concepts inherent in the understanding of these perspectives. This method may be considered as a guideline in business process and workflow modeling and also allows organizations to deal with the competitive pressure of the network economy and to improve the quality of service for the satisfaction of different stakeholders.

Index Terms— Business Process Modeling, Process abstraction, Workflows modeling, QoS, Customer satisfaction, Formal method.

---

1 Introduction

A business process typically refers to an enterprise process. Business process modeling (BPM) is one of the core methodologies developed to better represent the functional behavior of the information system dealing with the delivery of services to customers within an organization. Existing business process models are not based on formal approaches in the line of the numerous models such as abstract system, abstract integration, system abstraction and simulation and concrete system. In fact, it rather belongs to the family of informal UML-like models, which seriously limits its theoretical potential and leaves the door open for new researches [3, 7].

In the domains of business process and workflow modeling, a wide range of techniques have been widely used [2, 3, 13, 15]. Despite the popularity of some of them, there is no consensus on the modeling standards and concepts of business process and workflow in the delivery of services within an organization. However, there are many perspectives that need to be taken into consideration for a better management of workflows. These perspectives include process, organization, information, operation and the quality of service (QoS). The survey of business process and workflow modeling shows that most researches have been concentrated in the process perspective and neglected the other perspectives [13]. As consequence, enterprises are not fully productive and have many difficulties to deal with challenges of the network economy.

Focusing on these challenges, researchers have recognized the importance of knowledge in the productivity of organization [25]. Knowledge management allows enterprise to improve the quality of services offered to customers, increase customers' satisfaction, and reduce the cost of maintaining and managing services. Knowledge can be found and resides in various places within an organization and out of the organization. This knowledge represents experience, customer's needs related to services, customer perception of the quality of service, and other valuable management lessons, the functioning and the operation of the organization. Now, because knowledge management aims at the improvement of task processing [19], methodologies that aim at building knowledge management system have to examine the businesses within an enterprise. To this end, the modeling of business process and workflows within an organization must include knowledge abstraction. In the meantime, enterprises face the problem of capacity building of new staff. For the purpose of maintaining the QoS despite the retirement of some staff, human resource managers usually have to send new staff for training, and spend much money to this end. However, the result is not always satisfactory and the manager is obliged to team them with those who are experienced [27], so they can catch up. Sometimes, this is not possible due to the fact that, this action generally comes late with respect to the departure for retirement of some staff members. Within this time period, the enterprise pays double for the same workstations; this contributes to a reduction in the income. This situation can be overcome if knowledge concerning the processing procedures is stored for future use and knowledge transfer [19, 25]. Moreover, the process of moving along work-stations that comprises the processing chain, enables new employees to acquire a vast amount of tacit, as well as explicit knowledge. For this reason, it will be very difficult for any employee to present in detail the knowledge required by any work station in the accomplishment of a specific task. As a result, experienced employees sometimes will provide inaccurate information for knowledge transfer. Although individual employees will not always be accurate or remember specific information regard-

- *Atsa Etoundi Roger is currently a senior lecturer in the Department of Computer Science of the University of Yaounde I, Cameroon, and the Chief Information Officer in the Ministry of Public Service and Administrative Reform. E-mail:roger.atsa@uy1.uninet.cm*

ing the processing of certain tasks within an enterprise, an enormous amount of knowledge can be obtained from the experience of all the employees who have already handled the processing of such tasks. In order to do so, it is interesting to capture all knowledge concerning the processing of any tasks within an enterprise and store it for future use. The process that can facilitate the capturing and storage of this knowledge for later sharing is the process of managing knowledge [19]. In order to design a management tool within an enterprise, it will require the inclusion of knowledge management process in the design of all business processes and workflows that are supported within an organization. This is one of the challenges that various enterprises have to tackle in order to resist the increasing competitive pressure of global economy if they have to survive.

In this new economy, the difference between enterprises is not only based on the manner in which it tackles the above issues but also on the way they deal with their customers for the satisfaction of their needs. Organizations thus face unprecedented competition, forcing them to offer exceptional levels of service based on the desire of customers, whichever the sector of productive business process they find themselves [2, 5, 4, 6]. Thus, enterprises should make efforts in the improvement of the quality of service desired by customers. Customers with the same requirements must receive the same service undependably of the employees involved in the processing of associated tasks. Feedback from the customers should be obtained by the enterprise for the improvement of the quality of the service offered. In the daily life of an enterprise, customers come with new complains and problems, but also with ideas and praises. To deal with the satisfaction of customers within an enterprise, not only the quality of the final service or product has to be considered but also the quality of various outputs obtained from the execution of tasks should be considered since the final quality of service or product is an aggregation of intermediate quality of this artefact. The intermediate quality is the basis in definition of the final QoS. Therefore, QoS must be integrated in the modeling of a task and in the modeling of business process and workflow. This issue has not yet been tackled in the research field of BPWM.

This paper presents a method, so called ATSERO Method, which captures different abstractions of a business process in different levels within an enterprise. The approach is based on domain and requirement engineering [24,25], and formal methods namely Denotational Semantic [1,14]. The rest of the paper is organized as follows: section 2 presents part of the Denotational Semantics concepts suitable for our modeling approach, section 3 presents the proposed method for business process and workflows modeling, the last section 4 concludes the work and highlights some perspectives and future works.

2 Denotational Semantics Concepts

This section gives parts of Denotational Semantics [1,14] that are suitable in handling the modeling of business processes and workflows within an enterprise. The resulting mathematical machinery is important in modeling business process and workflow concepts. This domain of mathematical objects is defined in terms of partially ordered sets, least upper bounds, chains and continuity.

Definition 2.1 *(Partially Ordered Set)*

Let C be an arbitrary set. A partial order $\subseteq$ on C is a subset of $C \times C$ which satisfies the following for all c1, c2 and c3 in C:

1. $c \subseteq c$ (reflexivity),
2. if $c1 \subseteq c2$ and $c2 \subseteq c3$ then $c1 \subseteq c3$ (transitivity),
3. if $c1 \subseteq c2$ and $c2 \subseteq c1$ then $c1=c2$ (antisymmetry).

In this mathematical domain of objects, we are concerned not only with arbitrary sets with partial order, but also with sets of functions with their ordering. A partial ordering on a set of functions of type C1→ C2 can be derived from the orderings on C1 and C2.

Definition 2.2 *(Sequence)*

Let $(C, \subseteq)$ be a partially ordered set, $\langle c_0, c_1, \ldots \rangle$, also denoted by $\langle c_i \rangle_{i=0}^{\infty}$ is a sequence if and only if for all $i \in N$: $c_i \subseteq c_{i+1}$.

Definition 2.3 *(CPO)*

A complete partially ordered (CPO) set is a set C with a partial order $\subseteq$ which satisfies the following requirements:

1. there is a least element, denoted by $\perp$, with respect to $\subseteq$, i.e $\forall c \in C : \perp \subseteq c$,
2. each sequence $\langle c_i \rangle_{i=0}^{\infty}$ in C has a least upper bound in C.

Definition 2.4 *(Continuous Function)*

Let $(C_1, \subseteq_1)$, $(C_2, \subseteq_2)$ be two CPO's, the function $f : C_1 \rightarrow C_2$ is continuous if and only if for each chain $\langle c_i \rangle_{i=0}^{\infty}$ in C, the following holds: $f\left(\bigcup_{i=0}^{\infty} c_i\right) = \bigcup_{i=0}^{\infty} f(c_i)$.

Fact 1 *(Fixed Point Theorem)* let C be a CPO and $f : C \rightarrow C$, if f is continuous, then the least fixed point μf exists and is equal to $\bigcup_{i=0}^{\infty} f^i(\perp)$, where $f^0(\perp) = \perp$ and $f^{i+1}(\perp) = f\left(f^i(\perp)\right)$.

Definition 2.5 (*Least Fixed Point)*

Let $(C, \subseteq)$ be a CPO, $f : C \rightarrow C$ and let $x \in C$.

- x is a fixed point of f if $f(x)=x$
- x is a least fixed point of f if x is a fixed point of f and for each fixed point y of f, the relation $x \subseteq y$ holds

Definition 2.5 (*Partial ordering functions)*

Let $(C1, \subseteq_1)$, $(C2, \subseteq_2)$ be two partially ordered sets, *C1*→ *C2* the set of continuous functions, an ordering $\subseteq$ on *C1*→ *C2* is defined as follows, where $f, g \in C1 \rightarrow C2$: $f \subseteq g \Leftrightarrow \forall c \in C1 : f(c) \subseteq_2 g(c)$.

Definition 2.7 *(Least Upper Bound)*

Let $C' \subseteq C$. $c \in C$ is called a least upper bound of C' if:

1. c is an upper bound of C' i.e $\forall x \in C'$, $x \subseteq c$,
2. c is a minimal element of the set of upper bounds of C' i.e $\forall y \in C : ((\forall x \in C' : x \subseteq y) \Rightarrow c \subseteq y)$.

The least upper bound of a partially ordered set C' will be denoted by $\bigcup C'$.

Definition 2.8 *(Least upper Bound of a Sequence)*

The least upper bound of a sequence $\langle c_0, c_1, \ldots \rangle$ denoted by $\bigcup_{i=0}^{\infty} c_i$ or $\bigcup \langle c_i \rangle_{i=0}^{\infty}$ is defined as follows:

$\cup\langle c_i\rangle_{i=0}^{\infty}=\cup\{c|c\in\langle c_i\rangle_{i=0}^{\infty}\}$ where c in $\langle c_i\rangle_{i=0}^{\infty}$ means that c is an element of the sequence $\langle c_i\rangle_{i=0}^{\infty}$.

Fact 2 *(CPO of functions)* Let $(C_1,\subseteq_1)$ and $(C_2,\subseteq_2)$ be two CPO's, then $(C_1\to C_2,\subseteq)$ is a CPO.

Fact 3 *(Least upper bound of functions)* Let $(C_1,\subseteq_1)$ and $(C_2,\subseteq_2)$ be CPO's and let $\langle f_i\rangle_{i=0}^{\infty}$ be a chain of functions in *(C1→ C2)*, then the function $\lambda_{c_1}.\cup_{i=0}^{\infty}\langle f_i(c_1)\rangle$ is the least upper bound of this chain and therefore $\left(\cup_{i=0}^{\infty} f_i\right)(c_1)=\cup_{i=0}^{\infty} f_i(c_1)$ for all $c_1\in C_1$.
In the rest of this paper, μf denotes the least fixed point of f where f is a continuous function on a CPO.

3 The ATSERO Method

This section presents in an incremental manner a model of a business process within an enterprise. The definition of this model starts by the presentation of salient concepts that are suitable for the modeling of a business process and the associated workflows for the achievement of stakeholders needs according to the related degree of satisfaction.

3.1 The Environment Description Model

The environment is considered as a set of different metrics whose value may change [9, 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 analyse 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:

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

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. $+c\cap -c=\emptyset$

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*.

3.1.1 State of an Environment

A state is a snapshot of an environment within a time [10, 11, 12]. 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. In our work, we are not dealing with any kind of set of states, we are interested with *S* 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 $\cup_s$ known as a state where the goal of the business process is satisfied.

3.1.2 Knowledge Model

In [20], a goal oriented approach- for the definition of a business process requirement model, taking into account 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 in [20], 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. It can be shown formally that this approach exhaustively describes a business process. To do this, a formalism to model the requirements of a business have been is defined. The model is refinement of the one presented by Farida and Joel Brunet in [26]. The refinement is based on the formal definition of a knowledge bit or expressed requirement. A knowledge bit is defined as follows $\langle k, Ag, Ex, y, w, l, d, v\rangle$ 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.

3.1.3 QoS Model

The quality of service denoted by QoS represents the performances of the service which determine the projected level of satisfaction for the recipients of these 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 exist 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 a set of common criteria to evaluate the quality of service delivered within 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 3.1

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, denoted 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.

3.2 Task Description Model

A task is an atomic activity that cannot be split into smaller activities [16, 17]. 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 are defined a set of knowledge bits and quality of service to obtain after the execution of a task. A task is formally defined 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 expected 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*. 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 3.2 *(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 conditions. Based on the post condition of a task *t*, and the state *s* where *s (post(t))=true*, we conjecture that $t_s = +post(t) \cup -post(t)$.

Definition 3.3 *(Conflicting Tasks)*

The action of tasks within an environment can be conflicting since 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 the constraints defined in the equation (1) are satisfied:

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

Definition 3.4 *(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 $KB_{x1} = KB_{x2}$

Definition 3.5 *(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.*

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

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

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:

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

Definition 3.6 *(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.

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 3.1
Let *p* be an execution path and $t \in S \circ T(p(k))$ with $k \leq length(p)$ then there will always exist *m* such that *m*>k and *S(p(m))(post(t))=false*.

Lemma 3.2
Let *p* be an execution path then $S \circ T(p(length(p))) = \varnothing$.

Definition 3.7 *(State ordering)*
Let *P* be a path of length *n>1*, and $Sht_k = \langle S_k, SoT_k \rangle$ and $Sht_{k+1} = \langle S_{k+1}, SoT_{k+1} \rangle$ 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 3.3
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 defined modeling approach 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 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 3.8 *(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 $n_{t1} \leq n_{t2}$. 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 $t_1 \trianglerighteq t_2$ and $t_2 \trianglerighteq t_1$ 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*, $t_1 \trianglerighteq t_2$ and $t_2 \trianglerighteq t_3$ then $t_1 \trianglerighteq t_3$.

Lemma 3.4
The set of tasks *T* associated with the relation previously defined $\trianglerighteq$, i.e. $(T, \trianglerighteq)$, forms a complete partial ordered set.

3.2.1 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*→*S*>. The set of functions *S*→*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*. Hence, 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.

$$\forall o \in \Theta \left\{ \begin{array}{l} \exists t_1 \in T, o \in -post(t_1) \\ \exists t_2 \in T, o \in +post(t_2) \end{array} \right. or$$

Given a palette *P*, according to the environment changes within organizations and the different processing of tasks that can take place, different ways in which tasks can be executed have to be captured. SP_P is use to specify the set of processing paths that can be obtained from a palette *P*.

Lemma 3.5
Let *P* be a palette, $s \in S(P)$ a given state of the environment *E(P)* of *P*, there will always exist a processing path $p \in SP_P$ such that $s \in S(p)$, where $S(p)$ denotes the set of states of the path *p*.

Lemma 3.6

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 processing paths of ***T***, $ch(n)=\langle s_n, S \circ T_n \rangle$ such that $t \in S \circ T_n$.

3.3 Task processing

Based on the planning and scheduling of tasks processing 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 knowledge associated to these tasks. According to the context within which the performance of the tasks is taking place, 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, or that received from more experienced 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* changes 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*.

3.4 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 3.9 *(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. This is the main difference between the proposed 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 [15].

Lemma 3.7

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 3.8

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 3.10 *(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 3.11 *(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_{CH}^{th} state s_{lub} of *CH* satisfies *G* i.e. $s_{lub}=true$.

Definition 3.12 *(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 3.1

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.

3.5 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 proposed modeling approach 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 they 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, ∩ 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 ∩ 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 3.12 *(Human Actor)*

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

3.6 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 . 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 its 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 $bk(t,c) \subseteq kb_{em}(t, f_{em}(t))$.

3.7 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 3.13 *(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 $\omega_{tk}(tk)$ acquired 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.

In the proposed modeling approach, A workstation *wk* $=(Tks, KBs, \omega_{tk}, \omega_{pk})$ with an empty set of knowledge bits related to its given task is not accepted, 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 3.14 *(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, *WKs* 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 to.

Definition 3.15 (Strong well-Definedness staff)

Given an enterprise Org=*(Io,BPs,Emps,WFs,WKs,fewk)*, the associated employees are said to be strong well-definedness if and only if each employee *ag* appointed to a given work station has the required profile, that is if $ag=<Sk,Ex,\varepsilon,Dy,Id>\in Emps$ with $Ex=<TEx,KBs,\beta>$ *and* $fewk(ag)=wk$ *where wk* is a workstation $wk=(Twk,KBwk,\omega tk,\omega pk)$ then $Twk \subseteq TEx$, and *ag* has the necessary knowledge to process all the tasks that is, if $tk \in Twk$, then $\omega pk(tk) \subseteq \beta(tk)$.

Definition 3.16 (Weak well-Definedness Staff)

Given an enterprise Org=*(Io,BPs,Emps,WFs,WKs,fewk)*, the associated employees are said to be weakly well defined if and only if each employee *ag* appointed to a given workstation has the required profile, that is if $ag=<Sk,Ex,\varepsilon,Dy,Id>\in Emps$ with $Ex=<TEx,KBs,\beta>$ *and* $fewk(ag)=wk$ *where wk* is a workstation $wk=(Twk,KBwk,\omega tk,\omega pk)$ then $Twk \subseteq TEx$.

Definition 3.17 (Well organized workstation)

Given an enterprise Org=*(Io,BPs,Emps,WFs,WKs,fewk)*, the associated workstations are said to be well organized if and only if each workstation *wk*, there exits an employee *ag* appointed to this position, that is $wk = fewk(ag)$.

Definition 3.18 (Strong well-definedness enterprise)

Given an enterprise Org=*(Io,BPs,Emps,WFs, WKs,fewk)*, *Org* is said to be strongly well defined if and only if (i) the workstations are well organized, and (ii) the employees are strongly defined that is they all have the public and tacit knowledge required for their respective positions in the organization.

Definition 3.19 (Weak well-definedness enterprise)

Given an enterprise Org=*(Io,BPs,Emps,WFs,WKs,fewk)*, *Org* is said to be weak well defined if and only if (i) the workstations are well organized, and (ii) there exists at least one employee who has only the public knowledge required by his position.

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. Thus, 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 3.9 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 $\forall p = (C,V,fp) \in \phi$, $c \in C$, $f_{qmin}(c) \leq fp(c)$, and $\forall q = <C,V,fq> \in \phi$, $c \in C$, $fq(c) \leq f_{qmax}(c)$.

Conclusion

The main technical content of this paper is to present a novel methodology for the modelling of business processes and workflows within an enterprise. The model is defined formally using the denotational semantics. The defined methodology is based on the domain and requirement engineering. These two approaches enable the determination of salient concepts that are suitable for the abstraction of a problem. Hence, concepts like the environment, the context of execution of a task, the resources required for the performance of a task, the knowledge bit needed for the processing of a task by a human actor, and the quality of service expected by consumer stakeholders. These core concepts and others have been defined formally using the denotational semantics in order to model a business process, a workflow and enterprise.

This novel approach differs from the existing one as it does not explicitly define the dependency among tasks. Based on the task model, the relation between tasks is obtained straightforward. Another contribution of this work is the definition of a link between the processing of tasks and the quality of service. This was not obvious in the former approaches despite the fact that the quality of service is one of the factors that allows enterprises to be competitive and to deal with challenges of the network economy.

Moreover, based on the fact that, human actors are groping out of the process while others are entering, whereas those groping out, most of the time, have acquired knowledge in the processing of tasks, as a result enterprises are then forced to train the new actors in order to maintain the same level of the quality of the service. This requires additional expenditure and a decrease in productivity during the training period. In order to deal with this problem, the modeling of a business process should take into consideration knowledge suitable for tasks processing. The existing approaches for business process modeling have neglected this issue, and it becomes very difficult to integrate this requirement in the resulting model. The proposed ATSERO Method tackles this issue by proposing in the definition of a knowledge repository within an enterprise based on human actors in charge of tasks processing.

Based on the notion of chain, the processing path of a business process has been laid out. Each of these paths is required to reach the least upper bound state of the environment which satisfies the associated goal of a business process. This goal is defined as a set of criteria that are assigned for the quality of service evaluation. From the chain concept, the well-formedness and well definedness properties of a business process have been defined. These definitions are used to show that a business process is deadlock and livelock free. The modeling is completed by giving an abstract representation of an enterprise based on business processes and human actors who deal with the processing of associated tasks in order to deliver a given quality of service. The defined models can serve as a guideline for business process and workflow modeling within various organizations. In future works, more investigations based on case studies for practical issues shall be carried out, after which a support tool shall be developed.

REFERENCES

[1] Jesper Andersen,Ebbe Elsborg,Fritz Hengle, Jakob Grue Simonsen and Christian Stefansen, "Compositional specification of commercial contracts", Springer verlag, 2006

[2] Jorg Becker, Michael rosemann, and Chrisoph von Uthmann. "Business Process Management", Guideline of Business Process Modeling. Springer-Verlag Berlin Heidelbert, 2000

[3] F. Casati, S. Ceri, B. Pernici, and G. Pozzi. "Conceptual Modeling of Workflows". Springer Verlap, December 1995

[4] Amit P. sheth, Will van der Aalst, Ismailcem B. Arpinar. "Processes driving the networked economy". IEEE Concurrency, July-September 1999

[5] C.R.Ramakrishnan H. Davulcu, M. Kifer and I.V. Ramakrishnan. "Logic based modeling and analysis of workflows". ACM Sympodium on Principles of Database systems, pages 25–33, june 1998

[6] D. Hollinsworth. "The workflow reference model". Technical Report TC00-1003, Workflow Management Coalition, December 1994

[7] Tomasz Janowski and Ojo Adegboyega. "Formalising Feasibility and correctness of distributed business processes". Lecture Notes in Computer Science, number 2465, pages 432–443, 2002

[8] Daniel Krob. "Modelling of complex software systems: a reasoned overview". International conference on formal methods for networked and distributed systems, 2006

[9] Atsa Etoundi Roger Marthe Monessa, Marcel Fouda Ndjodo and Erick Zobo. "Feature-Oriented workflow modelling based on enterprise human resource planning", Business process management journal, 12(5):608–621, 2006

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

[11] Atsa Etoundi Roger. "A Domain Engineering Approach for Multi Perspective Business Process and Workflows Modeling", PhD thesis, University of Yaounde I, December 2004

[12] Atsa Etoundi Roger and Marel Fouda Ndjodo. "A Generic Abstract Model for Business Processes and Workflows Management", Bieter Gerald and Kirste Thomas, editors, 4th International Workshop on Mobile Computing, pages 62–72. IRB Verlag, Stuttgart Germany, 2003

[13] Oumaima Saidani and Selmin Nurcan. "Role-based approach modelling flexible business process", Business Process Modeling, Development and Support, 2006

[14] R.D. Tennent. "The denotational semantics of programming languages", Communication of the ACM, 1976

[15] W. van der Aalst and al. "Business process Management", Techniques for modeling Workflows and Their Support of Reuse, pages 1–15. Springer Verlag Berlin Heidelberg, 2000

[16] Atsa E. R., Fouda M. and Abessolo G. 2010. "A Denotational Semantics Methodology (DSM) Approach for Business Processes Modelling", International Journal of Computer Applications, Vol 1; N°1,2010

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

[18] D. Hollinsworth. The workflow reference model. Technical Report TC00-1003, Workflow Management Coalition, 1994

[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. International Journal of Computer Applications, Vol. 2 (12), 7350-7362, 2010

[20] R. Atsa Etoundi, M. Fouda Ndjodo, Christian Lopez Atouba. "A Goal Oriented Approach or the Definition of Business Process Requirement Model", International Journal of Computer Applications, Vol 9, N°7, 2010

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

[22] W.M.P. van der Aalst, K.M. van Hee, and G.J. Houben. "Modelling and analysing workflow using a Petri-net based approach", Proceedings of the second workshop on computer supported cooperative work, pages 31–50, 1995

[23] PMI Project Management Institute. A Guide to the Project Management Body of Knowledge (Pmbok Guide), Third Edition.Project Management Institute, 2004.

[24] Atsa Etoundi Roger, M. Fouda Ndjodo, Christian Lopez Atouba. "A Formal Approach for the inclusion of key performance Indicators in Business process Modeling", International Journal of Computer Applications, Vol 12; N°12; 2011

[25] IEEE Institute of Electrical Electronics Engineers. Guide to the Software Engineering Body of Knowledge. IEEE, 2004

[26] Farida Semmak, Joël Brunet, « Un métamodèle orienté buts pour spécifier les besoins d'un domaine », 23e Congrès INFORSID, pp115-132, mai 2005

[27] Atsa Etoundi Roger, M. Fouda Ndjodo, Priso Essawe Ndedi and Abessolo Alo'o Ghislain, "Improving the quality of service of a public workflow based on ant theory: A case study in Cameroon", EJISDC, Vol 41; N°1, 2010