Volume 2, No. 1, Jan-Feb 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Formal Ontological Framework for Business Process Management within a Virtual Enterprise for Customer Satisfaction

Roger Atsa Etoundi*
Department of Computer Science
Faculty of Science
University of Yaounde I
roger.atsa@uy1.uninet.cm

Fouda Ndjodo Marcel
Department of Computer Science
Faculty of Science
University of Yaounde I
marcel.fouda@uy1.uninet.c

Abstract: The competitive pressure of network economy obliges enterprises to share their (technological, procedural or human) resources in order to satisfy the requirements of customers. These requirements continue to increase the need to programmatically validate the outgoing products or services. This problem occurs when an enterprise does not have enough qualified resources to meet a given requirement for the satisfaction of its customers. External qualified resources coming from other enterprises are then required for its satisfaction. The main difficulty is how to assign a job to a resource solicited by other enterprises for the same purpose based on constraints such as time and availability. To tackle this problem, we design a generic ontological framework for the management of the execution of a workflow within a virtual enterprise. The methodology is based on the specification of requirements for efficient rooting of jobs to shared resources in a virtual enterprise by taking into consideration constraints such as the availability and the transportation cost. The RSL (*Raise Specification Language*) is used to carry out the modeling process.

Keywords: Virtual Enterprise, Virtual Business Process, Shared Resource Management, Quality of Service, Workflow Requirement Engineering

I. INTRODUCTION

The concept of workflow management has been around for less than years, but suitable systems have been available recently. This technology has many applications areas such as document management, enterprise resource management etc. Despite their popularities regarding the achievement of enterprise business goals, they are far from meeting the overall needs of enterprises. Among, their limitations, we can single out the scheduling problem [15] (e.g how to find an execution that obeys the constraints that embody the business logic of the workflow), and communication among these systems [16,3,14]. Research on workflow scheduling has largely concentrated on temporal constraints, which specify the syntactical correctness of a workflow [4,8]. The limitation of the scheduling problem in the workflow within an enterprise is extended in the virtual enterprise as each of the associated enterprises should be able to locally handle this issue. The management of shared resources within a virtual organization is a critical issue since it combines the scheduling and the communication problems. Since different parts of an organization develop their own workflow environments which, later, will have to communicate. However, these features are not fully supported by current workflow management systems [3,16]. Workflow architectures need to be designed with this in mind. It does not suffice to have a common model (like the one proposed by the Workflow Management Coalition [5]), since the problem involved is a much more complicated issue.

The goals of customers are not always met within an enterprise. This is most of the time due to resource limitation. To rigorously handle customers' requests, enterprises cooperate by sharing their resources. In this respect, they use internet to penetrate new markets, sharing supply chains in order to meet the challenge of the global market [16]. This integration has come out with new terms such as virtual business process and Virtual Enterprise concepts. A virtual business process of a virtual enterprise, also known as inter-organizational workflow [2], goes beyond a single enterprise boundary: it is constructed by combining the services different enterprises provide. To share resources, communication among enterprises is fundamental, but it is preeminent in virtual organizations. Without communication, the boundary spanning among virtual entities would not be possible. Given the burgeoning interest in this emerging phenomenon, it is surprising that very little empirical research exists on virtual organizations. Specially lacking are studies of communication processes within virtual organization settings.

A virtual organization is a collection of geographically distributed enterprises that are linked by electronic forms of communication and thus, rely on lateral, dynamic relationships for coordination. Despite its diffuse nature, a common identity holds the organization together in the minds of members, customers, or other constituents. In this paper, we describe a virtual organization as one that is depleted with external ties [6], managed via teams that are assembled and disassembled according to need [10,7], and consisting of employees who are physically dispersed from one another [17,9]. While workflow management systems have been proven to be successful in supporting intra-enterprise goals, a central coordination of independent enterprises cannot be achieved using similar means, as the autonomy of the participating systems should be preserved [1]. Therefore, collaboration cannot be guaranteed, as parties of virtual business goals are in conflict since they may require the same resources.

The execution of work items within a virtual enterprise is done by delegating them to resources of different enterprises. The fundamental problem is the communication between these enterprises to efficiently manage their shared resources in order to meet their respective business goals.

This communication will allow the enactment of the execution of workflow within the virtual enterprise. This will not only help in the management of the shared resources, but also in the execution of workflow as this latter depends on external event such as the availability of external resources. Thus, workflow management systems should be extended in order to model the interdependencies among enterprises and to control the coordination of tasks distributed over several shared resources. The rest of the paper is organized as follows: in section 2 various concepts required to model the virtual enterprise and to deal with the rooting of tasks to shared resources among enterprises are presented; in section 3 requirements that should satisfy the rooting process of tasks to shared resources are highlighted; section 4 deals with the effectiveness of tasks rooting to shared resources; in section 5, we consider the consistency of the entire information system of the virtual enterprise, while section 6 concludes the paper and highlights some future works.

II. THE SETTINGS

In this section, we will describe various concepts that will be used throughout this paper in order to capture salient requirements to design the inter-organizational transactions necessary to enact cooperation, coordination and control patterns governing the structure of transactions for the management of the shared resources (employees).

A. Definition 2.1 (Time)

Time denotes a specific point expressed in year, month, day, minute or second

We will not define time with all its granularities. The way it will be expressed in concrete is left open. We only require that time be partially ordered, and be equipped with the operation +. From two time instances, we define the time concept *Period* from which the concept *Duration* is defined.

type
Time, Duration,
Period' :: start: Time due : Time,
Period = {|p:Period', start(p) < due(p)|}
value
$<$: Time × Time → Bool,
$\leq$: Time ×Time → Bool,
$+$: Time × Duration → Time,
duration : Period → Duration

B. *Definition 2.2 (Business Process)*

A business process is defined by the abstract type *BP*. For each business process *bp*:*BP*, is associated a set of tasks *tasks(bp)* describing the process, the state *state(bp)* from which the execution of tasks will start, the goal *goal(bp)* to be achieved, the duration *duration(db)(t)* associated to each task $t \in$ *tasks(bp)* this relation is required to be different from the null duration captured by the value *null*, the dependency between tasks of *tasks(bp)* to be different from null) of each task and the ordering (*dependency*) between tasks, and the pre condition (*precond*)of each task. Some other concepts of business process may be found in [11,12].

type
BP, Task, Condition, Env, Place, State
value
tasks : BP → Task-set,
precond : BP → Task $\xrightarrow[m]{}$ Condition, where $\xrightarrow[m]{}$ denotes a map definition
goal: BP → Condition,
place :BP → Task $\xrightarrow[m]{}$ Place,
duration: BP → Task $\xrightarrow[m]{}$ Duration,
init : BP → State,
dependency: BP → Task $\xrightarrow[m]{}$ Task-set,
val :Condition × State → Bool

C. Definition 2.3 (Agent)

An agent (*RS*) denotes a resource capable of performing or participating in the execution of a job within an enterprise. However, we restrict ourselves to human resources (employees) in this paper. We require each agent to be associated with certain skills and an identifier (*RSID*).

type
RS,Skill,RSID
value
agent : RSID → RS,
skills : RS → Skill $\xrightarrow[m]{}$ Task-set
axiom
($\forall$ rs :RS, **dom** skills(rs) $\neq${}), ($\forall$rs:RS,s:Skill, s$\in$dom skills(rs) $\Rightarrow$ skills(rs)(s)$\neq${}) where **dom** denotes the domain of the map.

D. Definition 2.4 (Workflow)

A workflow (*WF*) defines an activity, the set of resources which will be involved in the achievement of tasks, the time constraint from which the execution may start or complete.

type
WF ::
bp:BP
starttime:TimePlus
deadline:TimePlus,
TimePlus == undt | time(construct:Time)
value
wf : WFID → WF, where WFID denotes the identification of a workflow.

E. Definition 2.5(Run)

Let *wf* denotes a workflow, a run of *wf* defines the way it may be executed.

Formally, a run is built from an abstract implementation of the associated business process which is a list of orthogonal tasks, and from the schedule of each task.

type
Implementation = (State $\xrightarrow[m]{}$ Task-set)*,
Run ::
implementation:Implementation
execution : Execution*,
Execution ::
exec : Task $\xrightarrow[m]{}$ RSID
execperiod : Task $\xrightarrow[m]{}$ Period
value
runs : WF → Run-set
axiom
($\forall$wf:WF, rn:Run, rn $\in$ runs(wf) $\Rightarrow$ordered_period(execution(rn)))

F. *Definition 2.6 (Diary)*

A diary defines an ordered sequence of no overlapped periods associated with a task. A diary is associated with functions which guarantee no overlapping of its periods.

type
Job ::
task : Task
period : Period,
Diary = Job-list
value
insertJob : Diary × Job $\xrightarrow{\sim}$ Diary

The function *insertJob* is partial meaning that a job cannot be inserted in a diary if its associated period overlaps with an existing period of the diary within which a task execution has been planned.

G. *Definition 2.7 (Enterprise)*

An enterprise (*EE*) describes an organizational entity with a non empty set of agents, the different activities from which different workflows to be executed are defined. Moreover, each enterprise (*ee*:*EE*) is associated with an identifier (*id:EEID*), and each agent within it is associated with an agenda within which its different schedules are kept.

type
EE,EEID
value
org : EEID → EE,
resources : EE → RS-set,
schedules : EE → RSID $\xrightarrow[m]{}$ Diary,
business : EE → BP-set,
share : EE → RS-set,
mobile : EE → RS-set,
locations : EE → RSID $\xrightarrow[m]{}$ Place ×Time,
cost : EE → Place ×Place $\xrightarrow[m]{}$ Duration,
planning : EE → WF-set

[a] *Lemma 2.1*

Let *e* be an enterprise, we require that each of its agent be associated with an agenda, that the agents sharing be part of the resources of *e*, two different enterprises be associated with two different identifiers, and that mobile agent [13] be associated with his current location.

axiom
(∀e:EE, **card** resources(e) = card **dom** schedules(e)),
(∀e:EE,rid:RSID , rid ∈schedules(e) ⇒agent(rid) ∈resources(e)),
(∀e:EE, share(e) ⊆ resources(e)),
(∀eeid,eeid':EEID, eeid ≠eeid' ⇒org(eeid) ≠org(eeid')),
(∀e:EE, card mobile(e) = card **dom** locations(e)),
(∀e:EE, rid:RSID , rid ∈**dom** locations(e) ⇒agent(rid) ∈ mobile(e)),
(∀e:EE, **dom** locations(e) ⊆ dom schedules(e))

H. *Definition 2.8 (Zone)*

Let *A*= $\{DB_1, ..., DB_n\}$ be a set of enterprises, *ZN* is said to be a domain of *A* if enterprises of *A* are connected among them via an intranet. Each domain is associated with an identifier, the different schedules of sharing agents and locations of the sharing agents which are mobile.

type
ZN,ZID
value
zone : ZID → ZN,
schedules : ZN → EEID $\xrightarrow[m]{}$ RSID $\xrightarrow[m]{}$ Diary,
locations : ZN → EEID $\xrightarrow[m]{}$ RSID $\xrightarrow[m]{}$ Place ×Time

[a] *Lemma 2.2*

Let *zid* and *zid'* be two different zone identifiers, then they are associated with two different zones. The schedules of agents in *org(dbid)* are the same in *zn*. All the mobile agents in the domain level are associated with thier current location and with their agenda.

axiom
(∀zid,zid':ZID , zid ≠zid' ⇒zone(zid) ≠zone(zid')),
(∀zn:ZN,dbid:EEID, dbid ∈dom schedules(zn) ⇒ schedules(zn)(dbid) = schedules(org(dbid))),
(∀zn:ZN,dbid:EEID, dbid ∈dom locations(zn) ⇒ dom locations(zn)(dbid) = dom locations(org(dbid))),
(∀zn:ZN, dom locations(zn) = dom schedules(zn)),
(∀zn:ZN,eid:EEID, eid ∈dom locations(zn) ⇒dom locations(zn)(eid) ⊆dom schedules(zn)(eid))

I. *Definition 2.9 (Global Domain)*

The set of domains interconnected via internet defines a global domain (*VE*).

type
VE
value
schedules : VE→ ZID $\xrightarrow[m]{}$ EEID $\xrightarrow[m]{}$ RSID $\xrightarrow[m]{}$ Diary,
locations : VE → ZID $\xrightarrow[m]{}$ EEID $\xrightarrow[m]{}$ RSID $\xrightarrow[m]{}$ Place ×Time

[a] *Lemma 2.3*

If *zi* and *ve* denote an identifier of a zone and a global domain respectively, and *zone(zi)* represents a zone of ve then we require that, regarding *zone(zi), ve* contains the same information defined in zone(zi).

axiom
(∀ve:VE,zi:ZID, zi ∈ dom schedules(ve) ⇒ schedules(ve)(zi)=schedules(zone(zi))),
(∀ve:VE,zi:ZID, zi ∈dom locations(ve) ⇒ locations(ve)(zi)=locations(zone(zi))),
(∀ve:VE, dom locations(ve) = dom schedules(ve))

III. SPECIFICATION OF CORRECTNESS REQUIREMENTS

The achievement of a business goal within a virtual enterprise is generally done by distributing tasks between agents of different enterprises. Defining if a given business goal will can be satisfied within a virtual environment is very complex as the resources may not always have enough time to execute tasks or be overloaded. In this section we identify a number of requirements that should satisfy the execution of a workflow within a virtual organization. When a workflow execution satisfies these requirements, the execution is said to be correct.

A. *Requirement III.A*

Customer goals should be met within the time constraints defined by the delivery of goods, services and the starting time of the execution of the business goal.

Value
run : WF ×VE → Run,

satisfy_tc : Run ×WF → Bool
axiom
(∀ve:VE,wf:WF,satisfy_tc(wf,ve) ⇒case startime(wf) of time(tm) →(∀t:Task ,is_in_run(t,wf) ∧ maximal(t,wf) ⇒tm ≤start(t,wf)) end),
(∀wf:WF,ve:VE, satisfy_tc(wf,ve) ⇒case deadline(wf) of time(tm) → (∀ t:Task, is_in_run(t,wf) ∧ minimal(t,wf) ∧cptime(t,wf) ≤ tm) end)
where is_in_run(t,wf) expresses the fact that t is in the run of wf, and start(t,wf) and cptime(t,wf) the starting and completion time of the execution of t, minimal(t,wf) expresses the fact that *t* is one of the first tasks to be executed in the achievement of *wf*, *maximal(t,wf)* expresses the fact that *t* is one of the last tasks to be executed in the achievement of *wf*.

B. Requirement III.B

All resources involved in the achievement of the business goal should be available. That is, they must be free within the time interval they have been assigned work items.
value
allfree : WF ×VE → Bool
allfree(wf,ve) ≡ (∀t:Task, task_of(t,wf) ⇒free_performer(t,wf,ve)),
where the term task_of(t,wf) defines the fact that t is a task of wf, free_performer(t,wf,ve) denotes the fact that the performer of t is idle during its execution.

C. Requirement III.C

All mobile resources involved in the achievement of *wf* have enough time to move to places where tasks are needed to be carried out.
value
enoughtime : WF ×VE → Bool
enoughtime(wf,ve) ≡(∀t:Task, task_of(t,wf) /\ assign_to_mobile(t,wf,ve) ⇒ movecost(t,wf,ve) ≤start(t,wf))
where assign_to_mobile(t,wf,ve) denotes the fact that *t* has been assigned to a mobile agent, and *movecost(t,wf,ve)* denotes the arrival time of the performer of *t* in the place of its execution.

D. Definition (Correct Execution in a VE)

Given a workflow *wf*, and a global domain *ve*, we say that wf is correct and we denote it by *correct(ve,wf)* if the above requirements are satisfied.
value
correct: WF ×VE →Bool
correct(wf,ve) ≡enoughtime(wf,ve) ∧ allfree(wf,ve) ∧satisfy_tc(wf,ve)

IV. INTERORGANIZATIONAL TRANSACTIONS

The basic cooperation mechanism among enterprises in a virtual enterprise is the process of supporting the management of shared resources. A shared resource may be solicited by more than one enterprise at a time. The main problem then is how to manage this type of resources in order to ensure that they are not assigned a large number of tasks to be executed within the same time interval. In this section, we define a model dealing with the efficient assignment of tasks to shared resources in a virtual organization.

Task assignment

When an enterprise *ee* deals with the execution of a workflow, the qualified resources to cope with this execution do not always exist in *ee*. In this case, such executions can be assigned to external resources.

A. Definition IV.A (Unachievable Task)

Let *t* be a task of the workflow whose execution is currently under planning in the enterprise *ee*, *t* is said to be unachievable in *ee* if there does not exist an agent in *ee* who has the skill or who is available to execute it.
value
unachievable : Task ×WF ×EE → Bool
unachievable(t,wf,ee) ≡ task_of(t,wf) ∧ (unable(t,ee) ∨ unavailable(t,wf,ee))
where the term unable(t,ee) expresses the fact that there is no agent qualified in ee to perform t while unavailable(t,wf,ee)expresses the fact that there does not exist agent available to execute t and to meet the deadline of wf.

When an enterprise fails to satisfy the goal of a given task, it needs to contact enterprises in order to receive a resource that can deal with the implementation of this task. The interaction between enterprises is based on two concepts request: (RQ) and response (RP), and it is done in three phases, first from between enterprise and the zone, then between the zone and the global domain, and finally from the global domain to the enterprise.

B. Definition IV.B (Request)

A request *rq* defines the data sent by an enterprise to the virtual organization in order to receive help.
type
RQ :: task:Task job:WFID org:EEID period:Period

C. Definition IV.C (Inbox)

Let *ee* and *ve* denote a zone and a global zone respectively, an inbox of *ee* or *ve* denotes the sequence of the requests received by *ee* or *ve*.
value
inbox : EE → RQ-set, inbox : VE → RQ-set, inbox : ZN → RQ-set

[a] Enterprise-Zone interaction}

When an enterprise fails to meet the goal of a task t, it sends a request to its zone manager who is in charge to check if there exists an enterprise with a qualified and available agent to execute *t*. The request *rq* sent by the enterprise is received by its zone *zn* when *rq* is inserted in the inbox of *zn*.
value
snd_request : EEID × ZN × RQ $\xrightarrow{\sim}$ ZN
axiom
(∀id:EEID,
zn:ZN,rq:RQ,unachievable(task(rq),wf(job(rq)),org(org(rq))) ∧
org(rq) ∈dom schedules(zn) ∧
case startime(wf(job(rq))) of time(tm) → tm ≤start(period(rq)) end ∧
case deadline(wf(job(rq))) of time(tm) → due(period(rq)) <= tm end =>
inbox(snd_request(id,zn,rq)) = inbox(zn) ∪{rq})

[b] Zone Manager

Requests received by a zone are processed in order to get the required resource or to transfer the request to the general manager of the overall enterprises. A request is transferred if it cannot be met in the zone level. A request is met in the zone if a resource has been defined for the achievement of the task, and there is sufficient time to move in order to execute the task if the agent is required to move in order to execute the task.

D. Definition 4.4

let *rq* be a request handling in the zone *zn*, the response (*RP*) from *rq* may be a success, that is, an agent has been found in the zone to deal with the execution of the task, or a failure (*und*), that is, an agent cannot be found in the zone level to perform the task of the request.

type

Assign:: org:EEID job:WF task:Task period:Period agent:RSID origine:EEID,

RP == und | allocate(response:Assign)

[a] Requirement 4.1

The decision support system for resource management in a virtual enterprise for decision making must provide a software component for checking the fulfillment of a request in the zone level.

value

get_rp : $ZN \times RQ \xrightarrow{\sim} RP$,

satisfy : $ZN \times RQ \rightarrow Bool$

satisfy(zn,rq) is rq = hd inbox(zn) $\wedge$ get_rp(zn,rq) ~= und $\wedge$ case get_rp(zn,rq) of allocate(rp) $\rightarrow$ ontime(rp,zn) $\wedge$qualified(rp) $\wedge$free(rp,zn) end where the term *ontime(rp,zn)* expresses the fact that the agent to whom the task has been assigned will arrive at the place of the execution of the task ontime, *qualified(rp)* means that the agent has the skill to perform the task, *free(rp,zn)* denotes that the agent is idle within the time interval within which the performance of the task will be done, and *get_rp(zn,rq)* defines the result of the request based on the information kept in the zone.

[b] Requirement 4.2

The decision support for the management of resources in a virtual enterprise for decision making must provide a repository in each enterprise to keep track of the External resource sharing the execution of its workflows.

value

share : $EE \rightarrow$ Assign-set

Definition 4.5 (Zone's notification) When the request has been met in the zone level, the zone sends the response to the enterprise. This process is called the notification of the zone.

value

notify : $ZN \times RQ \times EE \xrightarrow{\sim} EE$

axiom

($\forall$zn:ZN,rq:RQ,ee:EE,satisfy(zn,rq) => case get_rp(zn,rq) of allocate(a) $\rightarrow$

share(notify(zn,rq,ee))=share(ee) $\cup$ {a} end)

In the same manner, the enterprise within which the resource is permanently working is informed of the temporal assignment of task to one of its resources.

value

notify_owner : $ZN \times RQ \times EE \xrightarrow{\sim} EE$

axiom

($\forall$zn:ZN,rq:RQ,satisfy(zn,rq)=> let r=get_rp(zn,rq) **in** case r of allocate(a) $\rightarrow$ org(orogine(a)) = ee $\wedge$ share(notify_owner(zn,rq,ee))=share(ee) $\cup$ {a} end)

[c] Definition 4.3

In case the request is not met in the zone, the decision support for the management of resources in a virtual enterprise for decision making must provide a software component to shift this request to the global zone.

value

send : $VE \times RQ \times ZN \xrightarrow{\sim} VE$

axiom

($\forall$ve:VE,rq:RQ,zn:ZN,~satisfy(zn,rq)=>

inbox(send(ve,rq,zn))= inbox(ve) $\cup$ {rq})

The resource specified to deal with the execution of the task defined in the request can also be involved in the planning of another workflow. It is necessary to verify this situation for further decision making.

[d] Requirement 4.4

The decision support for the management of resources within a virtual enterprise for decision making must provide a software component to check if a resource has already been involved in the planning of a business goal in the virtual organization.

value

reserve : $ZN \times RQ \xrightarrow{\sim} Bool$

reserve(zn,rq) is satisfy(zn,rq) =>

case get_rp(zn,rq) of

allocate(a) $\rightarrow$ (exists a':Assign , a' $\in$ share(org(org(a))) => agent(a') = agent(a) $\wedge$ overlap(period(a),period(a'))) end where the term *overlap(period(a),period(a')))* denotes the fact that the two periods *period(a)* and *period(a')* overlap. If this is the case, it is necessary to look for another resource to deal with the execution of the same task. This will allow the manager of the enterprise to have another opportunity in the case the first the participation of the first resource in the execution of another workflow is confirmed.

[e] Requirement 4.5

The decision support for the management of shared resources within a virtual enterprise for decision making must provide a software component to define a resource available, if existed, to perform the execution of the task contained in the request.

value

lookup: $ZN \times RQ \xrightarrow{\sim} RP$

axiom

($\forall$zn:ZN,rq:RQ , satisfy(zn,rq) $\wedge$reserve(zn,rq) => case lookup(zn,rq) of $\rightarrow$ allocate(a)

get_rp(zn,rq)~=lookup(zn,rq) $\wedge$ notreserve(a) end)

where the term *notreserve(a)* expresses the fact that the resource *agent(a)* is not temporally overloaded, that is the period period(a) overlap with another period within which he may be required to perform a task. In the case another agent cannot be found in the zone level, the request is shifted to the global zone.

[f] Requirement 4.6

The decision support for the shared resources management of within a virtual enterprise for decision making must provide a repository to keep track of requests which require finding another available resource.

value

translookup : RQ × ZN × VE $\xrightarrow{\sim}$ VE,

lookupdata : VE → RQ-set

axiom

(∀zn:ZN,ve:VE,rq:RQ, satisfy(zn,rq) /\ reserve(zn,rq) ∧ lookup(zn,rq)=und => lookupdata(translookup(rq,zn,ve)) = lookupdata(ve) ∪{rq})

[i] 4.1.3 Global Zone Manager}

In the reception of the request coming from his/her zones, the global zone manager checks if the need of the request can be achieved based on the information in his/her possession.

[g] Requirement 4.7

To efficiently manage requests received in the global zone, the decision support system for decision making in the management of resources within a virtual enterprise should provide a software component to define the response associated to a request based on the information kept in the global zone server.

Value get_rp : VE × RQ $\xrightarrow{\sim}$ RP

Given a request received in the global zone, having defined the response associated to it, if the resource has been defined to deal with the execution of the task defined in the request, the global manager should send a notification to the enterprise manager.

[h] Requirement 4.8

The decision support system for the management of resources in a virtual enterprise for decision making should provide a software component to send a notification to the initiator of the request and to the enterprise within which the resource is working.

value

notify : VE × RQ × EE $\xrightarrow{\sim}$ EE,

notify_owner : VE × RQ × EE $\xrightarrow{\sim}$ EE

When notifying the enterprise which meets the objective request, the agent may already be involved in the planning process of another enterprise and has been assigned a work item such that the resulting time intervals overlap.

[i] Requirement 4.9

The decision support system for resource management in a virtual enterprise for decision making by the global zone manager must also provide a software component to check if the multiple temporally assignment of tasks to a resource are conflicting, and a function to find another resource for the performance of the task.

value

reserve : VE × RQ $\xrightarrow{\sim}$ Bool,

lookup : VE × RQ $\xrightarrow{\sim}$ RP

E. Enterprise Manager

When the enterprise has received the response of the request, there is a need for the enterprise manager to check if the task can be achieved.

[j] Requirement 4.10

The decision support system for resource management in a virtual enterprise for decision making by the enterprise manager must provide a software component to check if a task can be executed by an external resource.

type

Ack :: task:Task

value

ack : EE → Ack-set,

allocated : EE × Task → Bool

allocated(e,t) is (∃ ck :Ack,a:Assign , ck ∈ack(e) /\ a ∈share(e) ⇒task(ck)=t ∧task(a)=t),

get_rp : EE × WF × Task $\xrightarrow{\sim}$ Assign,

plan : EE × WF × Task $\xrightarrow{\sim}$ Plan

axiom

(∀e:EE,wf:WF,t:Task, t ∈tasks(bp(wf)) ∧allocated(e,t) ⇒ (∀a:Assign,a ∈ share(e)⇒ is_in(period(get_rp(e,wf,t)),period(a)))),

(∀e:EE,wf:WF,t:Task, t ∈tasks(bp(wf)) ∧allocated(e,t)=> plan(e,wf,t) = mk_Plan(agent(get_rp(e,wf,t)),period(get_rp(e,wf,t))))

where given two periods p and p', is_in(p,p') denotes that p is included in p'. When the enterprise has received the response from its request of resource, if the required resource has been found, it must be inserted in the run of the resulting workflow, else the goal pursued by the workflow cannot be achieved.

[k] Requirement 4.11

The decision support system for resource management in a virtual enterprise for decision making by the enterprise manager must provide a software component to update the plan associated to a workflow in the case an external resource has been found to execute its task.

value

insert_share : EE × WF × Task $\xrightarrow{\sim}$ EE

axiom

(∀e:EE,wf:WF,t:Task,t∈tasks(bp(wf))∧allocated(e,t)⇒insert_share(e,wf,t)= updaterun(e,wf,t,plan(e,wf,t)))

where the term *updaterun(e,wf,t,plan(e,wf,t))* denotes the insertion of the plan of the task *t* in the run of *wf*. If there is no resource available to deal with task, the workflow should be removed from the enterprise as the customer goal cannot be achieved.

[l] Requirement 4.12

The decision support system for resource management in a virtual enterprise for decision making by the enterprise manager must provide a software component to reject a workflow in the case an external resource cannot been found to execute its task, and to cancel all the reservation of the resources made.

value

reject : EE × WF × Task $\xrightarrow{\sim}$ EE,

cancel : EE × WF × Task × EE $\xrightarrow{\sim}$ EE

axiom

(∀e:EE,wf:WF,t:Task, t ∈tasks(bp(wf)) ∧~allocated(e,t) ⇒(∀a : Assign, a ∈share(e) ∩ share(org(origine(a))) ⇒ share(cancel(e,wf,t,org(origine(a))))= share(org(origine(a)))\{a}),

(∀e:EE,wf:WF,t:Task,t ∈tasks(bp(wf)) ∧~allocated(e,t) ⇒ planning(reject(e,wf,t)) = planning(e)\{wf})

When the workflow has been rejected from the enterprise, some resources of other enterprises reserved for the achievement of tasks of this workflow, are cancelled.

[m] Requirement 4.13

The decision support system for resource management in a virtual enterprise for decision making by the enterprise manager must provide a software component to cancel all the reservations made for a workflow whose goal cannot be met.

value

cancel : EEID × EE × WF $\xrightarrow{\sim}$ EE

axiom

(∀e:EE,wf:WF,t:Task, ~allocated(e,t) => (∀a:Assign, a ∈share(e)=>

share(cancel(origine(a),e,wf))=share(org(origine(a)))\{a}))

[n] Definition 4.6

A workflow within a virtual enterprise is satisfied if the goal of the associated business goal is satisfied, that is its goal can be achieved and all resources involved in its execution are available.

value

satisfy : VE × WF → Bool

satisfy(ve,wf) ≡ allavailable(ve,wf) ∧ontime(ve,wf) ∧

let impl = implementation(run(wf,ve)), g = goal(bp(wf)) in

case impl(len impl) of (s,tks) → val(g,s) ∧tks={} end end

where the term allavailable(ve,wf) denotes the fact that all the resources involved in the achievement of the goal of wf are all available, and ontime(ve,wf) denotes the fact that wf is achieved within its time constraints. When the goal G pursued by the workflow is satisfied, all the enterprises sharing the resources involved in the achievement of G and who may be overloaded must be discharged from the use of those resources within these time intervals.

[o] Requirement 4.14

The decision support system for resource management in a virtual enterprise for decision making by the enterprise manager must provide a software component to cancel the reservations of resources within time intervals these resources have been assigned tasks.

value

cancel : EE × WF × EEID $\xrightarrow{\sim}$ EE

axiom

(∀e:EE,wf:WF,a:Assign, satisfy(e,wf) ∧ a ∈external(e,wf)=>

(∀a':Assign, a' ∈ share(org(origine(a))) ∧

overlap(period(a),period(a)) =>

share(org(origine(a)))= share(cancel(e,wf,origine(a)))\{a'}))

where the term *external(e,wf)* defines the information of the external resources participating in the achievement of *wf.*

[p] Definition 4.7

The execution of a workflow *wf* by an enterprise *E* of a virtual enterprise is correct if the goal of the goal of *wf* is achieved and there exists external resources to deal with tasks whose performance cannot be done in *E*.

value

satisfy : EE × WF → Bool

satisfy(e,wf) ≡ achieve(e,wf) ∧(∀ri:RSID, ri ∈ exagent(e,wf) ⇒ (∃a:Assign , a ∈ share(e) ∧ agent(a) = ri)),

achieve : EE × WF → Bool

achieve(e,wf) is

let impl=implementation(planning(e)(wf)), g=goal(bp(wf))

in **case** impl(len impl) **of** (s,tks) → val(g,s) ∧tks={} end end

where *exagent(e,wf)* denotes the set of external resources involved in the execution of wf.

[q] Proposition 4.1 (Correct execution in VE)

A workflow in a virtual enterprise is said to be correct if the goal pursued by the workflow is satisfied meanning that the business goal is satisfied in the final state of the run, and all the resources are available to deal with work item within time constraints. That is:

(∀ve:VE, wf:WF,ei:EEID,zi:ZID, satisfy(org(ei),wf) ∧zi ∈ **dom** schedules(ve) ∧ ei ∈ **dom** schedules(ve)(zi) ⇒ satisfy(ve,wf))

The proof of this proposition is straightforward by using the definition of the function *snd_request*, and the function *satisfy* defining the satisfaction of the request in the zone or global zone which guarantee the properties of the satisfaction of *wf* in *ve*.

V. LOAD DISTRIBUTION}

Having defined how the plan of the execution of a workflow in a virtual enterprise is carried out, we need to guarantee the consistency of the information system of a whole system. This need requires distributing loads of resources involved in the execution of workflow whose goal is satisfied.

A. Requirement 5.1

The decision support system for resource management in a virtual enterprise for decision making must provide a repository to keep track of requests demanding to update *(the polymorphism to_update*) the loads of shared resources in the enterprise, zone and global zone.

value

to_update : EE → Assign-set,

to_update : ZN → Assign-set,

to_update : VE → Assign-set

B. Requirement 5.2

The decision support system for resource management in a virtual enterprise for decision making by the enterprise manager must provide a software component (the partial function *notify*) to notify enterprises whose resources are involved in the execution a workflow for their involvement in this process.

type

DomNotify = EE × WF × EEID × RID × Assign

value

notify : DomNotify $\xrightarrow{\sim}$ EE

axiom

(∀e:EE,wf:WF,ri:RSID,a:Assign, satisfy(e,wf) ∧ agent(a)=ri ∧job(a)=wf ∧a ∈ share(e) =>

to_update(notify(e,wf,org(origine(a)),ri,a))=

to_update(org(origine(a))) ∪ {a})

C. Requirement5.3

The decision support system for resource management in a virtual enterprise for decision making by the enterprise manager must provide a software component to update the diaries of resources after receiving the a request, and to send all requests received to update the diaries of its resources to its zone.

value

update : EE -~⟶ EE,

notify : EEID × ZN $\xrightarrow{\sim}$ ZN

axiom

(∀e:EE, to_update(db) ~= {} => (∀a : Assign , a ∈ to_update(e) ⇒resources(e)(agent(a))=
insertJob(mk_Job(task(a),
period(a)),resources(e)(agent(a))))),

(∀ ei:EEID,zn:ZN, to_update(org(ei)) ~={} ∧ei ∈dom zn =>

(∀a : Assign , a isin to_update(org(ei)) => to_update(zn)))

D. Requirement 5.4

The decision support system for resource management in a virtual enterprise for decision making by the enterprise manager must provide a software component (partial function *update*) to update the diaries of resources in the zone level while notifying (partial function *notify*) the global zone manager.

value

update : ZN $\xrightarrow{\sim}$ ZN,

notify : VE × ZN $\xrightarrow{\sim}$ VE

VI. CONCLUSION AND PERSPECTIVES

In this paper we have presented a model dealing with the management of a workflow in the virtual enterprise based on the efficient management of shared resources. Our approach was to describe an ontological framework for the management of shared resources in a virtual enterprise. The description is based on the configuration of the virtual enterprise which is decomposed into a set of enterprise zones. The request for resources for the achievement of business goal between enterprises within the same zone is carried out by sending the request to the manager of the zone who has the entire information regarding the associated shared resources. While the request for resource between enterprises from different zones is carried out via the zone of the sender of the request to the manager of the overall zones. Based on this protocol, we argue that the set of the requirements defined in throughout this paper can help to significantly improve the quality of service delivered to customers in this type of organization. Moreover, the proposed protocol can be seen as guideline in handling the problem of health care access in developing countries where the limitation of qualified human resources is a great challenge that these countries are trying to overcome in order to improve the health care system. This issue will be one of our future works in this area, which will be followed by a set of case studies.

VII. REFERENCES

[1] P. Bernstein. Middleware: A model for distributed system service. Communication of the ACM, pages 86-98, February 1996.

[2] Christoph Bussler. Enterprise-Wide Workflow Management. IEEE Concurrency, page 32, July-September 1999.

[3] Johann Eder, Michael Rabinovich, Heinz Pozewaunig, and Euthimios Panagos. Time Management in Workflow Systems.Lecture Notes in computer science, 1999.

[4] C.R.Ramakrishnan H. Davulca, M. Kifer and I.V. Ramakrishnan. Logic based modeling ad analysis of workflows. ACM sympodium on principles of Database systems, pages 25-33, june 1998.

[5] D. Hollinsworth. The workflow reference model. Technical report TC001003, Workflow Management Coalition,http://www.aiai.ed.ac.uk/WFMC/ , December 1994.

[6] Coyle J. and Schnarr N. The Soft-side challenges of the virtual corporation. Human Ressource Planning, 18:41-42, 1995.

[7] Lipnack J. and Stamps J. Virtual teams: Reaching Accross Space, Time and Organizations with Technology. NY: John Willey, 1997.

[8] V. Atluri N. Adams and W. Huang. Modeling and Analysis of workflow using Petri nets.Journal of Intelligent Information Systems, 10(2):131-158 march 1998.

[9] Barner R. The new millenium workplace: Seven changes that will challenge managers and workers. The futurist, 30:14-18, 1996.

[10] Grenier R. and Metes G. Going Virtual: Moving your Organisation into the 21th century.New Jersey: Printice Hall, 1995.

[11] Atsa E. Roger and Marcel Fouda Ndjodo. An Abstract Model for Workflows and Business Processes in CARI 2002, pages 239-247

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

[13] Atsa Etoundi Roger and Marcel Fouda Ndjodo. Mobile-Based Support for Business Processes: Feasibility and Correctness. In ACS-IEEE International Conference on Computer Systems and Applications, volume IEEE Catalog Number: 03EX722, ISBN: 0-7803-7983-7. IEEE, 2003.

[14] Atsa Etoundi Roger and Marcel Fouda Ndjodo. Security Based Approach to Data Consistency in a Virtual Enterprise. In ACS-IEEE International Conference on Computer Systems and Applications, volume IEEE Catalog Number: 03EX722, ISBN: 0-7803-7983-7. IEEE, 2003.

[15] Pinar Senkul, Michael Kifer, and Ismael H.Toroslu. A Logical Framework for Scheduling Workflow under Ressource Allocation Constraints. In The 28th VLDB conference, HongKong China, 2002.

[16] Amit P. Sheth , Wil van der Aalst, and Ismailcem B. Arpinar.Processes driving the networked economy. IEEE concurrency, page 18, July-September 1999.

[17] Clancy T. The latest world from thoughtful executives- the virtual corporation, telecommuting and the concept of team. Academic of Management Executive, 8(2):9-10, 1994.

