Dynamic Interconnection of Enterprise Workflow Processes

Karim Baïna, Khalid Benali, Claude Godart

Dynamic Interconnection of Enterprise Workflow Processes

Karim Baïna, Khalid Benali, and Claude Godart
LORIA - INRIA - CNRS (UMR 7503)
BP 239, F-54506 Vandœuvre-lès-Nancy Cedex, France
{baina,benali,godart}@loria.fr

ABSTRACT : Due to business process automation development, process interconnection becomes an important matter. Actually, process interconnection mechanisms are indispensable to co-ordinate business processes within and beyond organisation boundaries, aiming, for instance, to strength awareness inside virtual enterprises, to facilitate multinational e-transactions, etc. Therefore, thinking and proposing mechanisms to ensure interconnection between organisational business processes is becoming a hot research topic. Actually, existing business process modelling and enactment systems (workflow systems, project management tools, shared agendas, to do lists, etc.) have been mainly developed to suit enterprise internal needs. Thus most of these systems are not adapted to inter-enterprise co-operation. As we are interested in workflow process integration, we aim, through this paper, to provide a model supporting dynamic inter-enterprise workflow process interconnection. We consider the interconnection of enterprise workflow processes as the management of a "*workflow of workflows*" in which several heterogeneous workflow management systems (WFMS) coexist. This paper introduces our process interconnection model, its implementation, and its validation through an experimentation.

keywords : integration and interoperability of enterprise workflow applications, Internet-based inter-enterprise business process collaboration, co-operative information systems, business process wrappers, inter-organisational service integration, process interconnection contracts, out-sourcing based workflow interconnection, workflow systems, service exchange paradigm.

1 INTRODUCTION

Our purpose is to provide a framework to support dynamic enterprise workflow process interconnection. By *interconnection of enterprise workflow processes*, we mean the management of a "*workflow of workflows*" in which several heterogeneous workflow management systems will coexist. By *dynamics* of enterprise workflow process interconnection, we mean that process interconnection does not consider neither predetermined communication primitives, nor scheduled points of rendezvous. In other terms, an enterprise, aiming to interconnect its workflow process with another organisation workflow process (e.g. for out-sourcing of a piece of specific software development, for an online command of a service, for data exchange rendezvous in a virtual enterprise, etc.), has to discover and co-decide an interconnection contract at runtime. In fact, we have transformed the problem of interconnection of two workflow processes into the problem of dynamic out-sourcing between these processes. To be interconnected with other processes, a workflow process out-source dynamically parts of it to the other workflow processes. This enables interactions resulting from workflow interconnection to be limited in the time (*i.e.* to the out-sourcing period) and then to be well managed and controlled. Our paper's aim is to present our process service interconnection model, and is structured as follows : section 2 presents the process interconnection state of the art, section 3 formalises our process service interconnection model, section 4 presents an implementation of our model, and finally section 5 gives some hints on experimentation we made to validate our approach.

2 PROCESS INTERCONNECTION : PROBLEMS AND STATE OF THE ART

Due to business process automation development, process interconnection becomes an important matter. Although a wide spectrum of tools for process modelling and enactment exists (workflow systems, project management tools, shared agendas, to do lists, etc.), they have been developed to suit the intern needs of enterprises, and thus, are not adapted to inter-enterprise interconnection. Compared to other enterprise process systems, workflow processes are the most mature and operational. Meanwhile, they still have many drawbacks when considering enterprise process interconnection. In spite of WFMS normalisation efforts achieved by the W*f*MC (*Workflow Management Coalition*) ((WFMC 1996), (WFMC 1998), (WFMC 2000)), the OMG (*Object Management Group*) ((OMG 2000a), (OMG 2000b)) and the IETF (*Internet Engineering Task Force*) ((Whitehead and Wiggins 1998), (IETF 1999)), existing workflow management systems are :

- *heterogeneous* : considering their definition and execution environments (disparate syntax and semantics of business process definition languages -BPDL-, ad-hoc process instance management), and their access means (non standard compliant API) ;
- and *monolithic* : considering the absence or the poorness of their API, and the black box process instance encapsulation.

Because of heterogeneous and monolithic aspects of workflow management systems, developping generic models for enterprise workflow process interconnection is a big deal. In fact, there exist several approaches for interconnecting enterprise processes, among which, we highlight the six most important :

- *Process message oriented communication* :(Baker, Georgakopoulos, Schuster, Cassandra, and Cichocki 1999), (Casati and Discenza 2000), and BizTalk (Microsoft 2000) describe several techniques for workflow process communication through asynchronous typed message passing, with interest to adapt paradigms like subscribe-notify, push, pull to workflow processes ;
- *Process event synchronisation* : (Alonso, Agrawal, and Abbadi 1996), ICN (Ellis 1999), OPERA (Hagen and Alonso 1999), W*f*MC (WFMC 1996), and WF-nets (van der Aalst 1999) upgrade process message communication paradigms with event coordination languages and algebras for synchronising interleaving workflow processes ;
- *Process data and interface interoperability* : W*f*-XML (WFMC 2000), PIP (RosettaNet 2000), e-speak (HP 2001), and W*f*MC (Fischer 2000) establish interoperability frameworks for standardising workflow process data structures and interfaces.
- *Process data concurrency and access control* : (Alonso, Agrawal, and Abbadi 1996), (Edwards 1996), (Dewan and Shen 1998), and IETF WebDAV (IETF 1999) & SWAP (Bolcer and Kaiser 1999) go beyond simple data interoperability to control access within shared workflow process dataspaces ;
- *Process transactional exchange control* : ECOO (Godart, Perrin, and Skaf 1999), TRANSCOOP (Puustjärvi 1999), WIDE (Grefen 1999), WISE (Alonso, Hagen, and Lazcano 1999), and MQSeries (Leymann and Roller 2000) consider workflow process as advanced transactions, and propose transactional models forworkflow process execution and data sharing management ;
- *Process service exchange approach* : Service concept has been defined in many research fields : Object Oriented research (OMG 1997), Process Modelling research (Georgakopoulos, Schuster, Cichocki, and Baker 1999; Piccinelli 1998; Godart, Perrin, and Skaf 1999; Klingemann, Wasch, and Aberer 1999a; Casati, Ilnicki, Jin, and Shan 2000; Grefen, Aberer, Hoffner, and Ludwig 2000), Distributed System research (Kutvonen 1998; Benatallah, Dumas, Fauvet, and Rabhi 2003), etc. In the field of workflow research, CMI (Georgakopoulos, Schuster, Cichocki, and Baker 1999), OCoN (Giese and Wirtz 2000), Crossflow (Grefen, Aberer, Hoffner, and Ludwig 2000) & (Klingemann, Wasch, and Aberer 1999b), eFlow (Casati, Ilnicki, Jin, and Shan 2000), define process service contracts for workflow process interconnection. A process service can be seen as a software entity presenting process particularities and outcomes without totally reveaáling the process structure (*i.e.* its workflow implementation). A process service shows a functional abstraction of a process (or parts of a process) provided by an organisation. It specifies the amount of work that the organisation promises to carry out with a specific quality of service. It also specifies which parts of a workflow process it covers and how the requester could access to them. Proces service concept has been studied from several point of view : process service execution semantics abstraction (Georgakopoulos, Schuster, Cichocki, and Baker 1999), sub-workflow process service selection (Klingemann, Wasch, and Aberer 1999a), dynamic process service activities configuration (Casati, Ilnicki, Jin, and Shan 2000), process service control flow level abstraction (Grefen, Aberer, Hoffner, and Ludwig 2000), service methods and events wrapping (Benatallah, Medjahed, Boughettaya, Elmagarmid, and Beard 2000), etc. Process service structure is to be seen as a co-operation pattern that relevantly supports dynamic workflow process interconnection and cooperation behaviours.

Compared to other approaches, *process service exchange approach* supports enterprise cooperation modelling in a very effective way. Actually, by it forces of abstracting enterprise workflow processes to be interconnected, process services are the *most adapted to build high level models for enterprise cooperation and generic models independent of workflow process particularities*. Moreover, process service exchange

approach offers a high level paradigm which is very open to extensions dealing with other approches basic paradigms (*e.g.* communication : message passing, data interoperability ; coordination : event synchronisation ; execution control : data access control, transaction management, etc.).
Hence, to build our dynamic enterprise workflow process model, we have chosen the process service exchange approach.

Our process service interconnection model will enable co-operating enterprises to structure, classify, and compare process services, to select dynamically a provided process service among those matching a required process service, and finally to keep possible their business processes co-operating through process service wrapping. This process service wrapping will realize the out-sourcing based interconnection of each workflow process interconnected couples. A process service may concern either long e-transactions (e.g. outsourcing the development of pieces of software, subscription to full e-learning sessions, etc.), or short e-transactions (e.g. online book commands, enactment of administrative processes, data exchange rendezvous in a virtual enterprise, etc.).

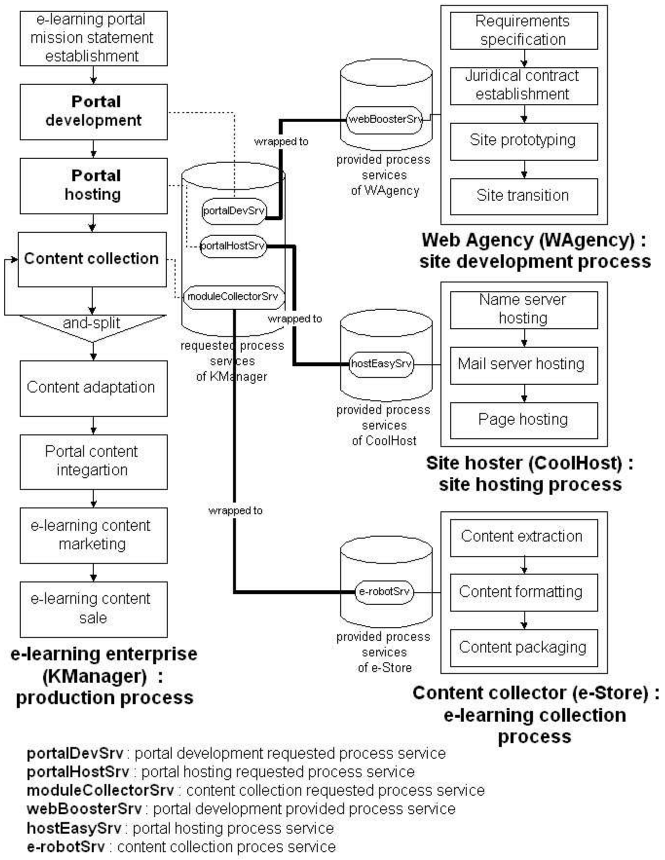
e-learning portal mission statement establishment
Portal development
Portal hosting
Content collection
and-split
Content adaptation
Portal content integartion
e-learning content marketing
e-learning content sale
e-learning enterprise (KManager) : production process
portalDevSrv
portalHostSrv
moduleCollectorSrv
requested process services of KManager
wrapped to
wrapped to
wrapped to
webBoosterSrv
provided process services of WAgency
hostEasySrv
provided process services of CoolHost
e-robotSrv
provided process services of e-Store
Requirements specification
Juridical contract establishment
Site prototyping
Site transition
Web Agency (WAgency) : site development process
Name server hosting
Mail server hosting
Page hosting
Site hoster (CoolHost) : site hosting process
Content extraction
Content formatting
Content packaging
Content collector (e-Store) : e-learning collection process
portalDevSrv : portal development requested process service
portalHostSrv : portal hosting requested process service
moduleCollectorSrv : content collection requested process service
webBoosterSrv : portal development provided process service
hostEasySrv : portal hosting process service
e-robotSrv : content collection proces service

Figure 1 – *An e-learning interconnection example*

Example To illustrate our approach, let us consider the following example within an e-learning context (figure 1). It is inspired from a real case of e-learning enterprise (Baïna, Baïna, Baïna, Baïna, Humbert, and Humbert 2002). Let KManager (an knowledge management and e-learning enterprise) be a service requester enterprise. Let WAgency (a web agency enterprise), CoolHost (a site hosting enterprise), and e-Store (an e-learning content collection enterprise) be three (among other) service provider enterprises. On the one hand, KManager requires three types of services : a portal development service (portalDevSrv), a portal hosting service (portalHostSrv), and an e-learning content collection service (moduleCollectorSrv). Such requested services can be implemented by several provided services. On the other hand, WAgency proposes an Internet site development process service (webBoosterSrv), CoolHost proposes an Internet site hosting process service (hostEasySrv), and e-Store proposes an e-learning content construction process service (e-robotSrv). After process service matching and negotiation sessions, KManager chooses WAgency with its process service webBoosterSrv which matches portalDevSrv, CoolHost with its process service hostEasySrv which matches portalHostSrv, and e-Store with its process service e-robotSrv which matches moduleCollectorSrv.

3 OUR PROCESS SERVICE INTERCONNECTION MODEL

The modelling of process service interconnection is based on a metamodel describing our service oriented approach, on structures participating to our enterprise process service interconnection model and on facilities presenting the dynamics of our model and its operational aspects

3.1 *Process Service Approach : Meta Model*

To tackle enterprise process interconnection problems, our approach considers three abstraction levels (or three layers) : workflow layer, process layer, and process service layer (c.f. figure 2).

Figure 2 shows that our oriented process service approach aims to present enterprise workflow processes, evolving inside monolithic and heterogeneous workflow management systems, as processes able to be interconnected through process services. This interconnection yields an inter-enterprise process that represents "*a workflow of workflows*" whose management is distributed on all interconnected entreprises as follows :

- *processes :*
  - Each enterprise possesses several processes ;
  - A process is composed of several process activities.
- *workflows :*
  - Each process delegates the execution of its

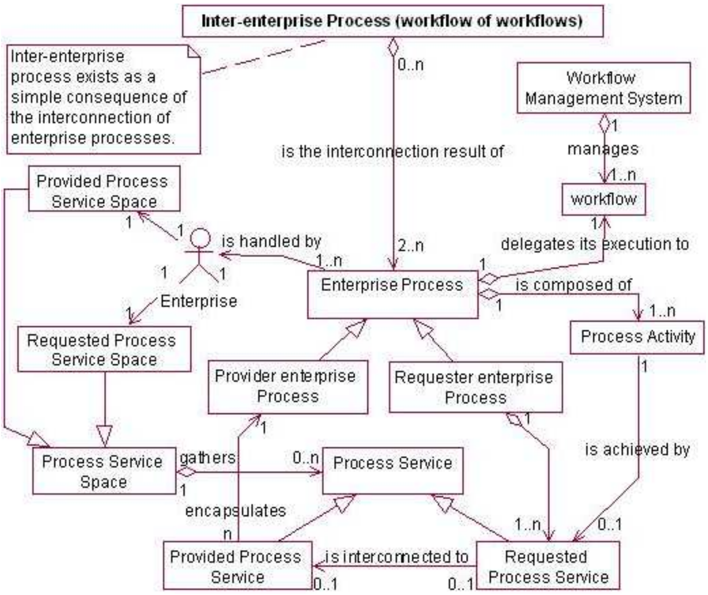
Inter-enterprise Process (workflow of workflows)
Inter-enterprise process exists as a simple consequence of the interconnection of enterprise processes.
0..n
Workflow Management System
1
manages
1..n
workflow
is the interconnection result of
Provided Process Service Space
1
1
is handled by
1..n
2..n
delegates its execution to
1
1
1
Enterprise
1
Enterprise Process
is composed of
1
1..n
Requested Process Service Space
Process Activity
Provider enterprise Process
Requester enterprise Process
1
1
1
Process Service Space
gathers
0..n
Process Service
is achieved by
1
encapsulates
n
1..n
0..1
Provided Process Service
is interconnected to
Requested Process Service
0..1
0..1

Figure 2 – *Interconnection Approach : Meta Model*

activities to a workflow ;

  - The management of a workflow is specific to its WFMS engine (workflow management system).

- *process services :*
  - A enterprise process activity can be achived by the entreprise means or by an external service. We call a service achieving a process activity a process service ;
  - Each enterprise possesses a *requested process service space* and a *provided process service space* ;
  - A requested process service describes needs to accomplish a process activity ;
  - A provided process service encapsulates a process that represents an ability to achieve a process activity ;
  - Process interconnection is done through the wrapping of their requested and provided process services.

Figure 3 describes enterprise collaboration within our process service oriented approach. The operation <publish>represents the *publishing* (requesting and providing) of a process service, the operation <find> represents the *discovery* of process service, while the operation <negotiate>represents a process service parameters *negotiation*. Finally, the operation <bind> represents the dynamic *interconnection* between an abstract process service (a requested process service) and a concrete process service offer (a provided process service). Beside the fact that our approach supports negotiation facility (which is not ensured by classical service approaches), its strengthens these approaches with symmetric aspects of process service publishing, discovery, negotiation and interconnection.

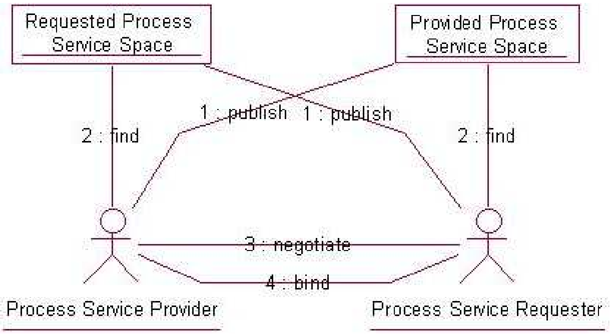
Requested Process Service Space
Provided Process Service Space
1 : publish
1 : publish
2 : find
2 : find
3 : negotiate
4 : bind
Process Service Provider
Process Service Requester

Figure 3 – *Interconnection Approach : Collaborations*

3.2 Process Service Interconnection model : Structures

3.2.1 Process Structure

Our process interconnection model is initially based on the F. Leymann and D. Roller process definition model (Leymann and Roller 2000). If this model can be applied very well for traditional workflows (within one enterprise), it does not consider explicitly process interconnection. Our objective is to enrich this model with new concepts and definitions in order to support some process interconnection aspects. Most of studied interconnection process models focus on process control flow and data flow definition without caring about two important process access points : process instance methods and process instance notification events.

The UML class diagram of the figure 4 defines the enterprise process structure as follows :

– a process is defined by a process graph and a process interface ;

– a process graph describes the process controlflow structure (Leymann and Roller 2000). It is the composition of nodes (process activities), edges (process activities transitions) and conditions (transition guards defined by business rules predicates) ;

– a process interface (or API) is the composition of methods (gathers all process instance reading and updating methods) and events (process event notification that are triggered off from a process instance during its execution). Process interface depends on the process definition and is not common to all processes.

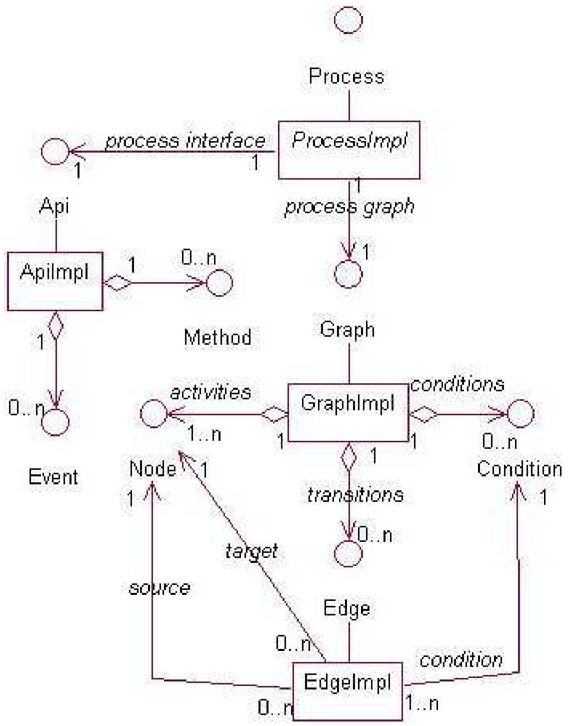

Figure 4 – *Process Structure*

Example Let us present a process example (figure 5) :

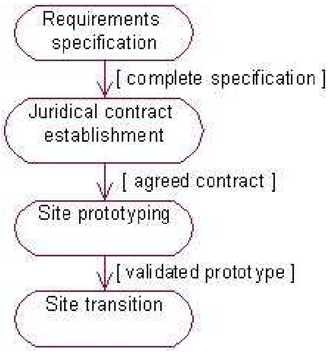
Requirements specification
[complete specification]
Juridical contract establishment
[agreed contract]
Site prototyping
[validated prototype]
Site transition

Figure 5 – WAgency *Process Graph*

WAgency *process instance method*	
m_1	fetchProcessInstanceState(..)
m_2	fetchActivityInstanceState(..)
m_3	fetchWorkItem(..)
m_4	getProcessInstanceAttributeValue(..)
m_5	fetchWorkItemAttribute(..)
m_6	getWorkItem(..)
m_7	fetchActivityInstanceAttribute(..)
m_8	fetchActivityInstance(..)
m_9	getActivityInstance(..)
m_{10}	getWorkItemAttributeValue(..)
m_{11}	getActivityInstanceAttributeValue(..)
m_{12}	fetchProcessInstanceAttribute(..)

WAgency *process instance event*	
e_1	TerminatedProcessInstanceNotification
e_2	StartedProcessInstanceNotification
e_3	TerminatedActivityInstanceNotification
e_4	StartedActivityInstanceNotification
e_5	AvailableNewDataNotification

3.2.2 *Process Service Structure*

The UML class diagram of the figure 6 defines the process service structure as follows :

- a process service is defined as a specific process wrapper that has a category, a profile and a visibility contract ;
- a process service category determines the process object type and its classification ;
- a process service profile describes a relational structure defining a set of process named-typed-values attributes ;
- a process service visibility contract represents a subset of the wrapped process interface. Otherwise, process service visibility contract "hides" process interface.

Example Let us present a process service example : WAgency.WebBoosterSrv process service structure

(*category* = e_learning_portal_development,
name = "WAgency.WebBoosterSrv",
profile = (duration=3 (month),
price=100 (Keuro),
dynamic_sites = true,
dbms = "MySQL",
XML_use = true,
Java_use = true,
JSP_use = true,
flash_use = false),
process = site_development_process)

WAgency.WebBoosterSrv *instance visibility contract method*	
m_1	fetchProcessInstanceState(..)
m_4	getProcessInstanceAttributeValue(..)
m_{11}	getActivityInstanceAttributeValue(..)
m_{12}	fetchProcessInstanceAttribute(..)

WAgency.WebBoosterSrv *instance visibility contract event*	
e_1	TerminatedProcessInstanceNotification
e_2	StartedProcessInstanceNotification
e_5	AvailableNewDataNotification

3.2.3 *Process Interconnection through Process Services*

The W*f*MC has established a well known problem of *process interconnection by nested sub-process model*. The W*f*MC *nested sub-process model* expresses

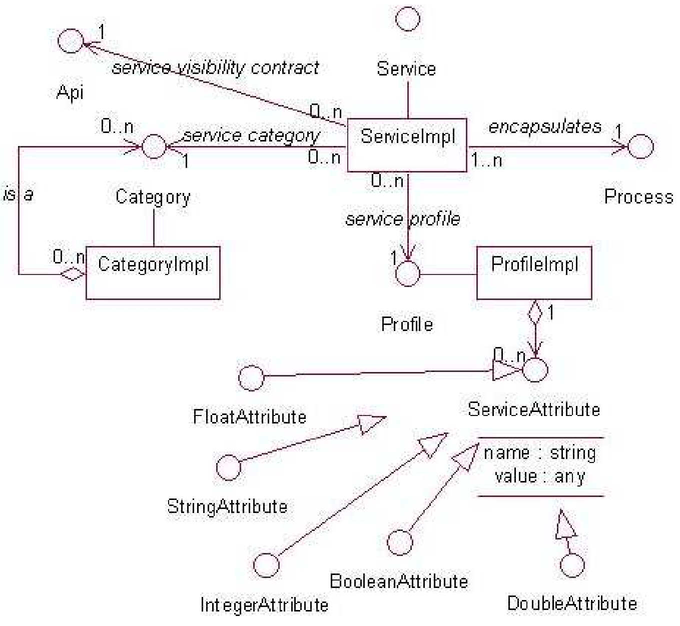

Figure 6 – *Process Service Structure*

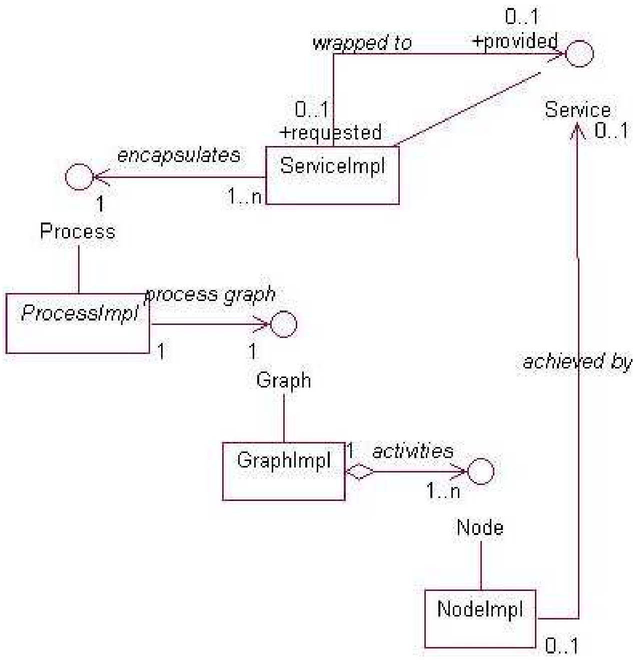

Figure 7 – *Process Interconnection through Services*

that an instance of an activity A_{i_j} belonging to a process P_i enacts remotely a known instance of an other process P_k and waits for its completion (WFMC 1996).

Moreover, the W*f*MC has defined eight levels of process interoperability. The coexistence is the second lowest interoperability level (among these eight levels). Coexistent process interconnection are processes that do not possess any common interoperability standard. They meanwhile share, the same environment -machine or operating system or network- to be able to manage and achieve parts of the same process (Fischer 2000).

Process interconnection through process services aims to resolve the problem of *coexistent process interconnection through a dynamic variant of nested subprocess model*. That means that an instance of an activity A_{i_j} of a process P_i discovers dynamically a process P_k that suits its realisation profile, adapts it, wraps to it, instantiates it, enacts it dynamically, cooperates with it, and waits for its completion.

Thus a process service structures will be the glue keeping possible this dynamic coexistent process interconnection as shown in the UML class diagram of the figure 7. Our complete approach is detailed in (Baïna 2003).

3.3 Process Service Interconnection model : Dynamics

Process service structure is a wrapper that represents a functional and semantic abstraction of a process. It enables the classification, the indexing, the comparison, and the discovery of a certain type of process. This supposes that enterprises, within each business community, agreed about common process service language (e.g. business key concept ontologies, business service taxonomies,. . .) to define and understand process services. Research and normalisation work are still emergent in this promising field (UDDI.Org 2000; W3C 2001; Microsoft 2000; UNCEFACT and OASIS 2000).

The dynamics of process service interconnection model will be presented through its facilities of publishing, discovering, negotiating and interconnectiong process services.

3.3.1 Process Service Spaces

The UML class diagram of the figure 8 defines the process service spaces structure as follows :

- a process service space is a set composed of process services ;
- an enterprise possesses four types of process service spaces :
  - private process service space : gathering all process services that the enterprise creates and keeps private to other enterprises before its publishing ;
  - requested process service space : accessible process service space gathering all process services that the enterprise requests (expresses the need of outsourcing to an external enterprise). Each requested process service knows its requester enterprise ;
  - provided process service space : accessible process service space gathering all process services that the enterprise can achieve by

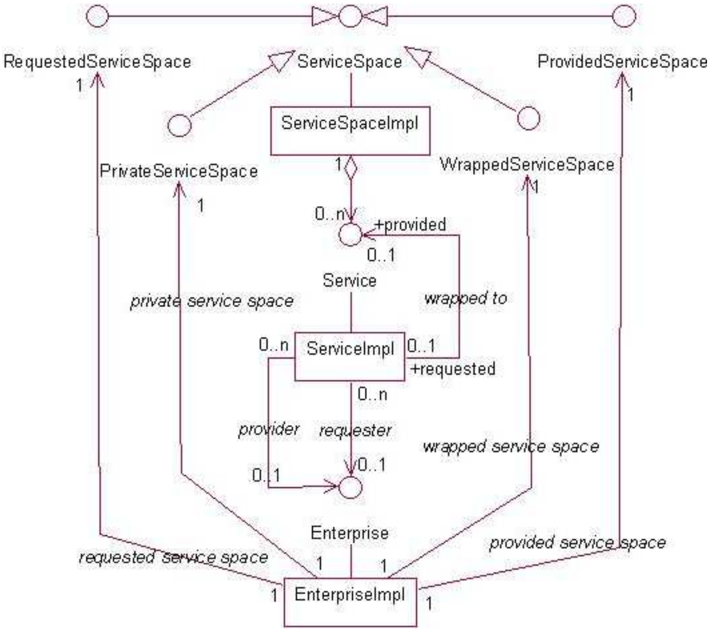

Figure 8 – *Enterprise Process Service Spaces*

her own means (expresses the capability to handle a requested process service of an external enterprise). Each provided process service knows its provider enterprise ;

– and wrapped process service space : accessible process service space gathering all process services that have been already wrapped (their requesting or providing expression has been satisfied by an external process service). Each wrapped process service knows both its requester and provider enterprise.

3.3.2 Process Service Publishing

Publishing of a process service deals with the communication of its description to other enterprises in order to find common agreement about enterprise process interconnections. Publishing of a process service is the result of the following steps :

1. The enterprise creates its process service within its private process space ;
2. the enterprise solicits its private process service space to publish (request of provide) its created process service ;
3. the private process service space constructs a clone of the process service and adds this clone to the requested process service space or to the provided process service space depending of the publishing type.

Actually, every subscribed enterprise possesses views on other enterprise requested and provided process service spaces. Publishing events notify all subscribed enterprises of the updating of their views. These views permits to every enterprise to discover, to negotiate, and to be interconnected through other enterprise published process services.

3.3.3 Process Service Discovery

Process service discovery deals with the application of algorithms that enable the evaluation and comparison of process services in order to help process service requesters (or providers) to find the process services that match their requested (or provided) process services in the best way. Discovery of a process service (for instance a requested one) is the result of the following steps :

1. The enterprise creates and provides a process service we will name “provided” ;
2. for ($set\ :=\ \emptyset$, i := 1; i <= n; i++) loop
   (a) the enterprise retrieves descriptions requested process services published in the provided service space of the enterprise e_i : $services_i\ :=\ \{rs_{i1}, rs_{i2}, \ldots\}$;
   (b) $set\ :=\ set\ \cup\ services_i$;

   end loop ;
3. the enterprise computes a neighbourhood algorithm on the process service provided among the set of possible requested process services set. This algorithm iterates on each element of set to find out the requested process services that suit the needs of the process service “provided”. This neighbourhood algorithm is based on distances and matching measures that could be instantiated according to the application ;
4. The discovery finishes by building a set of requested process services $\{nrs_1, nrs_2, \ldots\}$ that are neighbour to the process service “provided”.

Process service discovery is symmetric, it means that process service requesters can also discover provided process services that suit their needs. More details about matching and distance measures can be found in (Baïna, Benali, and Godart 2001).

Actually, views of subscribed enterprise possesses on other enterprise requested and provided process service spaces are then organised using discovery measures that permits them to browse a restricted computed projection of the wide process service space.

3.3.4 Process Service Negotiation

Process service negotiation enables to decide dynamically and to adapt all interconnection parameters between processes to be interconnected (e.g. profile, visibility contract, process graph accessibility, etc.).

Negotiation of a process service is the result of the following steps :

1. *solicitation phase :* the client contacts primarily

the server and expresses its negotiation request (e.g. a new process service profile, a new process service visibility contract, etc.) ;

2. *effective negotiation phase :* the client and the server exchange messages to build the set of acceptable solutions. An exchange protocol handles turn taking rules for expressing negotiation acts ;
3. *selection phase :* finally, the client or the server choose a solution among those expressed during negotiation phase. The server take its decisions according to the selected solution (e.g. allocating a solicited resource, tuning access rights for some service, etc.).

Actually, each subscribed enterprise is not only able to browse its views on other enterprise requested and provided process service spaces. However, it can also dynamically change these views by negotiating with process services publishers to customise their services for suiting its workflow process out-sourcing needs or offers.

More details about generic negotiation component can be found in (Munier, Baïna, and Benali 2000).

3.4 Process Service interconnection model in action : Protocol

Our process service interconnection protocol enables enterprise workflow processes to be interconnected and keep possible their cooperation by the following four steps :

1. *workflow processes definition :* every workflow process is defined as a graph that manages process execution model within an enterprise chosen workflow management system ;
2. *workflow processes adaptation :*
   - *workflow process service provider adaptation :* a workflow process providing services of a certain category has to be associated to a program or an instance of a class that implements the ConcreteProcessInterface corresponding to this category. This adaptation has to be written in a language supported or inter-operable with the WFMS ;
   - *workflow process service requester adaptation :* a workflow process requesting services has to express that its activities that need to be outsourced (interface Node) are achieved by external process services. This adaptation has to program, in a language supported by the WFMS, the enactment of the requested process services and the cooperation protocol that will be used to interact.
3. *dynamic workflow processes interconnection by process services :*
   (a) *process service definition :* process services are created by requester and provider enterprises within their private process service spaces. Process services are defined by their name, textual description, category, profile, encapsulated process (if the process service is to be provided) and visibility contract that desires (if it is to be requested) or that authorises (if it is to be provided).

   (b) *process service publishing :* process services are published by enterprises as provided or requested in their respective accessible requested or provided process service spaces ;

   (c) *process service discovering :* process services publisher can look for process services that suit the needs of its process service definition ;

   (d) *process service negotiation :* process services publishers can negotiate with each others their process service requested and provided definitions (profile and visibility contract) to agree on common satisfying process service definition ;

   (e) *process service wrapping :* process services wrapping deals with committing and dispatching agreed process service definitions (profile and visibility contract) on both process service views (the requester view and the provider view).
4. *workflow processes cooperation through process services :* workflow processes can begin their cooperation through process services that adapt them. This cooperation between wrapped process services can vary from method invocation or event passing (according to agreed visibility contract), to data exchange or synchronisation on process execution states. workflow processes cooperation through process services is to be considered as a generic paradigm that admits a wide panel of process cooperation modes.

4 IMPLEMENTATION

Our dynamic workflow process interconnection model has been implemented within our co-operative environment *DISCOBOLE* (DIStributed CO-operation and Business prOcess on LinE). *DISCOBOLE* integrates process and process service structures with their manipulation algorithms as innovative *CORBA application objects* gathered inside an *process interconnection and cooperation facilities*. Through these facilities *DISCOBOLE* supplies basic mechanisms for process interconnection and cooperation applications.

DISCOBOLE is implemented in Java on the CORBA broker architecture JacORB (Brose and Noffke 2002).

- *Process graph :* Beside interfaces and classes previously introduced, the UML class diagram of the figure 9 introduces the interface WorkflowEngine that represents a WFMS. A process graph and its components (interfaces Graph, Edge, Node and their realization classes) are implemented by a workflow inside a particular WFMS ;

- *Process Interface :* a process interface is represented by an IDL interface specific to the process application domain. This concrete interface extends the interface Process and specifies the new process category through its fields, methods and events. Process interface is schematized, in the UML class diagram of the figure 9, by the interface ConcreteProcessInterface and its realization class ;

- *Provider workflow adaptation :* both interfaces ConcreteProcessInterface, ConcreteProcess and their realization class schematize provider workflow adaptation. To provide a process service of a certain category, a workflow provider has to be adapted by a specific concrete process interface implementation.

- *Process meta-information (reflective extraction) :* the interface Process possesses, among others, two operations extract_profile() and extract_api() that extract respectively the profile and the API of a process service that is able to encapsulate (*i.e.* to be provided by) a process. We have chosen Java language reflection to implement both operations. For instance, this permits to dynamically explore an implementation of the interface Process, instantiate it by calling dynamically one of its constructors, etc.

- *Process service category, profile, and visibility contract :* the process service category (class CategoryImpl), profile (class ProfileImpl), and visibility contract (class ApiImpl) handle extracted process meta-information on its interface, relations with other interfaces, typed attributes, methods and events;

- *Process service :* a process service (class ServiceImpl) handles process meta-information then makes possible the process service publishing, discovery, negotiation, and process interconnection, and controlled access. A process service is related to a process proxy and to the process adapter (Gamma, Helm, Johnson, and Vlissides 1994). A process service is related to *process proxy* because, it controls dynamically the access to the process it encapsulates (process method invocation and process event notification rights). However, a process service cannot be assimilated to a process proxy because it does not supply complete or restricted process interface, it also offers mechanisms for discovery, negotiation and interconnection.

  Moreover, a process service is related to *process adapter* because, it procures a new interface to the process that it adapts (e.g. discovery, negotiation, interconnection). Additionally, a process service is able to dynamically extract the category and the attributes defining the process it adapts in order to build a profile enabling process discovery and comparison with other adapted processes. Meanwhile, a process service cannot assimilated to a process adapter because it ensures additionally process access control according to negotiation results.

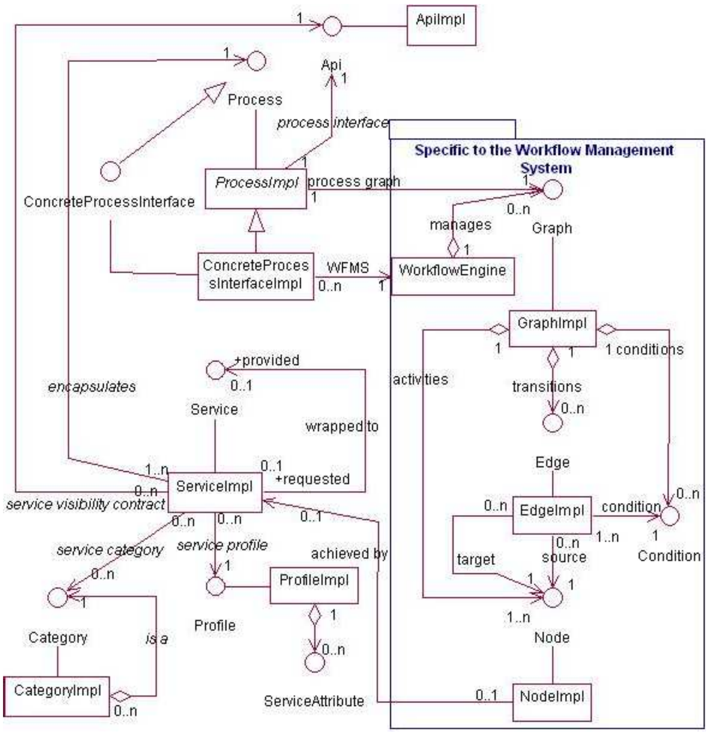

Figure 9 – *Interconnection Model Implementation*

5 EXPERIMENTATION

To experiment our enterprise workflow process interconnection model, we have deployed our e-learning enterprise context using *DISCOBOLE*. Each of the four enterprises KManager, e-Store, WAgency, and CoolHost uses a workflow management system to manage its business processes. *DISCOBOLE* is the environment that will enable them to interconnect dynamically these processes.

As shown in figure 10, in our experimentation, we selected and used three heterogeneous WFMS to model enterprise business processes : a lightweight component based WFMS *Breeze* (DSTC 2002), an object oriented PetriNets WFMS *Renew* (Kummer, Wienberg, and Duvigneau 2001), and a W*f*MC compliant WFMS *WorkCoordinator* (or WCO) (Hitachi 2002; Baïna, Charoy, Godart, Grigori, el Hadri, Skaf, Akifuji, Sakaguchi, Seki, and Yoshioka 2002). These three WFMS are written in different languages and evolve in different environments. In order to keep possible the interconnection of their defined workflow processes, adaptation is achieved according to the WFMS supported language and environment (c.f. section 3.4). Breeze workflow processes adaptation has been programmed in Java on a CORBA architecture (ORBacus). While Renew workflow processes adaptation, it has been programmed in Java on the same *DISCOBOLE* CORBA architecture (JacORB). Finally, for WorkCoordinator workflow processes, the adaptation has been programmed in C++ on WorkCoordinator CORBA architecture (VisiBroker).

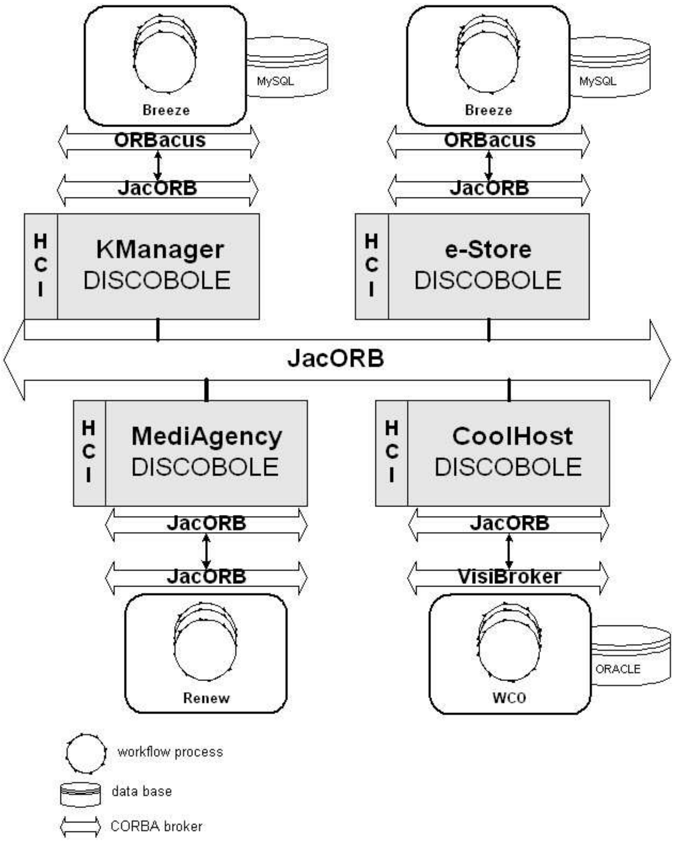

Figure 10 – *Example DISCOBOLE-Deployment*

6 CONCLUSION AND PERSPECTIVES

Our paper is a contribution in enterprise workflow interconnection domain which is a hot reseach topic as far as the current B2B (Business to Business) boom is concerned. In spite of normalisation efforts, WFMS are still presenting monolithic and heterogeneous drawbacks. Thanks to processes and process services frameworks, our model bypasses these drawbacks by enabling enterprise workflow processes interconnection within a "*workflow of workflows*" in which several workflow management systems coexist. Our model has been developed to support a wide panel of workflow management systems and experimented to prove the realisability of dynamic enterprise workflow processes.

ACKNOWLEDGEMENTS

We would like to thank W. Gaaloul and S. Baïna for their development participation within the project *DISCOBOLE*. Moreover, we would like to thank Dr. F. A. Rabhi from School of Information Systems, UNSW, Sydney New South Wales, Australia as well as the anonymous reviewers of this paper.

REFERENCES

Alonso, G., D. Agrawal, and A. E. Abbadi (1996, October). Process Synchronisation in Workflow Management Systems. In *8th IEEE Symposium on Parallel and Distributed Processing (SPDS'97)*, New Orleans, Louisiana.

Alonso, G., C. Hagen, and A. Lazcano (1999, June). Process in Electronic Commerce. In *ICDS workshop on Electronic Commerce and Web-Based Applications*, Austin, Texas, USA.

Baker, D., D. Georgakopoulos, H. Schuster, A. Cassandra, and A. Cichocki (1999, September 2-4). Providing Customized Process and Situation Awarness in the Collaboration Management Infrastructure. In *4th IFCIS Int. Conf. on Cooperative Information Systems (CoopIS'99)*, Edinburgh, Scotland, pp. 79–91. IEEE Computer Society Press.

Baïna, K. (2003). *Un Modèle Orienté Services Procédés pour l'Interconnexion et la Coopération des Procédés d'Entreprises*. Computer PhD Thesis, Université Henri Poincaré - Nancy 1, LORIA, France.

Baïna, K., J. Baïna, S. Baïna, S. Baïna, C. Humbert, and J.-C. Humbert (2002, June 6-7). bio-expert, Knowledge Management Platform and Experiment for Bio-Medical e-Learning. In *4th International Workshop on Enterprise Networking and Computing in Health Care Industry, Technically co-sponsored by IEEE (HealthCom'02)*, Nancy, France, pp. 138–141.

Baïna, K., K. Benali, and C. Godart (2001, September 5-7). A process service model for dynamic enterprise process interconnection. In C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella (Eds.), *9th Int. Conf. on Cooperative Information Systems, In Cooperation with*

IFCIS (CoopIS'01), Number 2172 in LNCS, Trento, Italy, pp. 239–254. Springer-Verlag.

Baïna, K., F. Charoy, C. Godart, D. Grigori, S. el Hadri, H. Skaf, S. Akifuji, T. Sakaguchi, Y. Seki, and M. Yoshioka (2002, May 1-3). CORVETTE: A Cooperative Workflow Development Experiment. In L. M. Camarinha-Matos (Ed.), *3rd IFIP Working Conference on Collaborative Business Ecosystems and Virtual Enterprises (PRO-VE'02)*, Sesimbra, Portugal, pp. 169–180. Kluwer Academic Publishers.

Benatallah, B., M. Dumas, M.-C. Fauvet, and F. A. Rabhi (2003). *Patterns and Skeletons for Parallel and Distributed Computing*, Chapter Towards Patterns of Web Services Composition, pp. 265–296. Springer-Verlag.

Benatallah, B., B. Medjahed, A. Boughettaya, A. Elmagarmid, and J. Beard (2000, September 14-15). Composing and Maintaining Web-based Virtual Enterprises. In *1st Workshop on Technologies for E-Services, In Cooperation with VLDB'2000 (TES'00)*, Cairo, Egypt.

Bolcer, G. A. and G. Kaiser (1999, January-February). SWAP: Leveraging the Web To manage Workflow (SWAP). In *IEEE Internet Computing*. computer.org/internet/: IEEE Computer Society.

Brose, G. and N. Noffke (2002). *JacORB, www.jacorb.org*.

Casati, F. and A. Discenza (2000, March). Supporting Workflow Cooperation Within and Across Organisations. In *15th ACM Symposium on Applied Computing (SAC'00)*, Como, Italy, pp. 19–21.

Casati, F., S. Ilnicki, L. J. Jin, and M. C. Shan (2000, June 8-9). eFlow: an Open, Flexible, and Configurable Approach to Service Composition. In *2nd International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems (WECWIS'00)*, Milpitas, California, pp. 125–132.

Dewan, P. and H.-H. Shen (1998). Flexible meta access-control for collaborative applications. In *Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW'98)*, Primitives for Building Flexibile Groupware Systems, pp. 247–256. ACM Press.

DSTC (2002, February 15). *Breeze : workflow with ease*. Australia, www.dstc.edu.au/Downloads/: DSTC (Distributed Systems Technology Centre).

Edwards, W. K. (1996, November). Policies and roles in collaborative applications. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, Boston, Massachusetts, USA, pp. 11–20. ACM Press.

Ellis, C. A. (1999). *Computer Supported Cooperative Work*, Chapter Workflow Technology. John Wiley and Sons.

Fischer, L. (Ed.) (2000, October). *The Workflow Handbook 2001*. Published in association with the Workflow Management Coalition (WfMC).

Gamma, E., R. Helm, R. Johnson, and J. Vlissides (1994). *Design Patterns, Elements of Reusable Object-Oriented Software*. Massachusetts: Addition-Wesley.

Georgakopoulos, D., H. Schuster, A. Cichocki, and D. Baker (1999). Managing Process And Service Fusion In Virtual Enterprises. *Information Systems, Special Issue on Information Systems Support for Electronic Commerce 24*(6).

Giese, H. and G. Wirtz (2000). The OCoN Approach for Object-Oriented Distributed Software Systems Modeling. In *Software Engineering and Petri Nets, Workshop within the 21st International Conference on Application and Theory of Petri Nets, Aarhus, Denmark, June 26*.

Godart, C., O. Perrin, and H. Skaf (1999, March 23-24). COO : a workflow operator to improve cooperative modelling in virtual processes. In *9th International Workshop on Research Issues on Data Engineering : Information Technology For Virtual Enterprises (RIDE-VE'99), Sponsored by the IEEE Computer Society*, Sydney, Australia.

Grefen, P. (1999). Advanced Architectures for Transactional Workflows or Advanced Transactions in Workflow Architectures. In *International Process Technology Workshop (IPTW'99)*.

Grefen, P., K. Aberer, Y. Hoffner, and H. Ludwig (2000). CrossFlow: cross-organisational workflow management in dynamic virtual enterprises. In *International Journal of Computer Systems, Science and Engineering (IJCSSE'00)*, pp. 277–290.

Hagen, C. and G. Alonso (1999, May/June). Beyond the Black Box: Event-based Inter-Process Communication in Process Support Systems. In *19th International Conference on Distributed Computing Systems (ICDCS'99)*, Austin, Texas, USA.

Hitachi (2002). *WorkCoordinator Workflow System*. www.hitachi.co.jp/Prod/comp/soft1/wco/: Hitachi Ltd.

HP (2001, January). *Architectural Specification, Release A.0, www.e-speak.net*. www.e-speak.net: Hewlett Packard.

IETF (1999, February). *HTTP Extensions for Distributed Authoring - WEBDAV*. www.ics.uci.edu/ webdav: IETF (Internet Enegineering Task Force), Network Working Group.

Klingemann, J., J. Wasch, and K. Aberer (1999a, June 14-18). Adaptative outsourcing in cross organizational workflows. In *11th International Conf. On advanced Information Systems Engineering (CaiSE'99)*, Heidelberg, Germany.

Klingemann, J., J. Wasch, and K. Aberer (1999b). Deriving service models in cross organizational workflows. In *9th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (ITVE'99)*, Sydney, Australia. IEEE Computer Society Press.

Kummer, O., F. Wienberg, and M. Duvigneau (2001, July 3). *Renew - User Guide*. Germany, www.renew.de: University of Hamburg, Department for Informatics, Theoretical Foundations Group, Distributed Systems Group.

Kutvonen, L. (1998). *Trading services in open distributed environments*. Computer PhD Thesis, Department of Computer Science, University of Helsinki, Finland.

Leymann, F. and D. Roller (2000). *Production Workflow, Concepts and Techniques*. Prentice-Hall, Inc.

Microsoft (2000, December). *Microsoft BizTalk Server : BizTalk Framework 2.0: Document and Message Specification*. www.biztalk.org.

Munier, M., K. Baïna, and K. Benali (2000, September 6-8). A Negotiation Model for CSCW. In O. Etzion and P. Scheuermann (Eds.), *5th IFCIS Int. Conf. on Cooperative Information Systems, In Cooperation with VLDB 2000 (CoopIS'00)*, Number 1901 in LNCS, Eilat, Israel, pp. 224–235. Springer-Verlag.

OMG (1997, December 5). *Autonomous Decentralized Service System (ADSS) Domain Special Interest Group Whitepaper ver 1.0 (ads/97-12-01)*. www.omg.org: OMG (Object Management Group).

OMG (2000a, February 14). *Workflow Management Facility Convenience Document combining dtc/99-07-05 dtc/2000-02-03 (WF RTF 1.3 Report)*. www.omg.org: OMG (Object Management Group).

OMG (2000b, April). *Workflow Management Facility Specification, V 1.2*. ww.omg.org: OMG (Object Management Group).

Piccinelli, G. (1998, August 20-22). Distributed Workflow Management: The TEAM Model. In *3rd IFCIS Int. Conf. on Cooperative Information Systems (CoopIS'98), Sponsored by IFCIS, The Intn'l Foundation on Cooperative Information Systems*, New York City, New York, USA, pp. 292–299. IEEE-CS Press.

Puustjärvi, J. (1999). *Transactional Workflows*. Ph. D. thesis, Department of Computer Science, University of Helsinki, Finland.

RosettaNet (2000, January 4). *Partner Interface Process (PIP) Release 1.3*. www.rosettanet.org.

UDDI.Org (2000, September). *Universal Description, Discovery and Integration (UDDI) Technical White Paper*. www.uddi.org.

UNCEFACT and OASIS (2000, October 17). *ebXML Technical Architecture Specification*. www.ebXML.org.

van der Aalst, W. M. P. (1999). Interorganizational workflows: An approach based on message sequence charts and Petri nets, System Analysis and Modeling, 34(3):335-367.

W3C (2001, March). *Web Service Description Language (WSDL) version 1.1*. www.w3.org/TR/wsdl: W3C (World Wide Web Consortium).

WFMC (1996, October 20). *Workflow Standard - Interoperability, Abstract Specification, WFMC-TC-1012, V. 1.0*. www.wfmc.org: WFMC (Workflow Management Coalition).

WFMC (1998, July). *Workflow Management Application Programming Interface (Interface 2 and 3) Specification, WFMC-TC-1009, V. 2.0*. www.wfmc.org: WFMC (Workflow Management Coalition).

WFMC (2000, May 1). *Workflow Standard - Interoperability, Wf-XML Binding, WFMC-TC-1023, V. 1.0*. www.wfmc.org: WFMC (Workflow Management Coalition).

Whitehead, E. J. and M. Wiggins (1998, September-October). WebDAV: IETF Standard for Collaborative Authoring on the Web. *Software Engineering*.