INFORMATICA, 2009, Vol. 20, No. 3, 323–342

An Agent-Based Model for Autonomic Coordination of Inter-Organizational Business Processes

Eric ANDONOFF, Wassim BOUAZIZ, Chihab HANACHI,
IRIT, Université de Toulouse, 2 Rue de Doyen Gabriel Marty
31042 Toulouse, Cédex 9, France
e-mail: {andonoff, wassim.bouaziz, hanachi}@univ-tlse1.fr

Lotfi BOUZGUENDA
MIRACL, Institut Supérieur d'Informatique et du Multimédia, Route de Tunis
BP 242, 3021 Sakeit Ezzit, Sfax, Tunisie
e-mail: lotfi.bouzguenda@ isimsf.rnu.tn

Received: October 2008; accepted: March 2009

Abstract. Inter-Organizational Workflow (IOW) aims at supporting the collaboration between several autonomous and heterogeneous business processes, distributed over different enterprises or organizations. Coordination of these processes is a fundamental issue that has been mainly addressed in a static context, but it still remains open in a dynamic one such as the Internet in which IOW applications are more and more enacted nowadays. In such a context, Multi-Agent Systems (MAS) are known to be a natural solution for modeling IOW since they provide adequate abstractions and specific mediators to cope with IOW coordination. Consequently, this paper provides an agent-based model for coordinating business processes involved in a dynamic IOW. This model is a triplet (E, M, R). E is the set of coordinated entities. It corresponds to the different business processes that may be published, discovered or deployed by IOW partners. M is the media supporting coordination. It is a multi-agent architecture compliant with the Workflow Management Coalition architecture and integrating specific components devoted to coordination issues. Finally, R is the set of rules governing the coordination. In our context, R is described through an organizational model aiming at structuring the interaction among the coordinated entities and the different components of the architecture.
Keywords: inter-organizational workflow, business process, dynamic coordination, agent-based model.

1. Introduction

Inter-Organizational Workflow (IOW) is a key technology to help enterprises/organizations to face the dynamic and open worldwide economic market in which they are involved. These organizations put in common their respective business processes in order to produce a new service, which is a value-added service that they are quite unable to produce individually. In such a context, IOW is an adequate technology since it supports the cooperation of distributed and heterogeneous business processes running in different enterprises/organizations (van der Aalst, 1999).

A fundamental problem for IOW is the coordination of the different business processes involved in it. By coordination, we mean all the work needed to put all these processes together in order to provide the global common goal in an efficient manner.

This coordination problem has been deeply investigated in a static context (van der Aalst, 1999; Chebbi *et al.*, 2006) but it remains open in a dynamic (or loose, see Divitini *et al.*, 2001) one in which the different partners (enterprises/organizations) of the IOW are not necessarily known before or during its execution either because they are unknown at design-time or no more available at run-time.

Moreover this dynamic context is more and more required giving the maturity of Internet standards (SOAP, WSDL, UDDI and BPEL for instance) for enterprises/organizations cooperation. Consequently, the coordination problem introduced before may be stated as follows: how to autonomically maintain the coordination of different business processes involved in an IOW? How to face the unavailability at run-time or the not knowledge at design-time of an organization responsible for the execution of a business process involved in an IOW? By autonomic we mean that the different business processes automatically and autonomously coordinate their activities during the execution of a dynamic IOW. In conclusion, autonomic coordination is now a topical issue, which needs to be tackled and solved.

As introduced in Bouzguenda *et al.* (2008), autonomic coordination in dynamic IOW raises several specific problems:

- Finding partners, which consists in, for a requester organization, selecting one or several provider organizations able to execute a requested business process.
- Negotiation of a business process between the requester organization and the previous selected provider organizations. Negotiation criteria may be as varied as due time, quality of the business process, visibility of its evolution and way of executing it. The result of this step is the identification of the provider organization in charge of the requested business process.
- Contract signature between the requester and the selected provider aiming at defining execution rules of the requested business process.
- Distributed execution of the business process and its monitoring.

The problem being addressed in this paper is "how to provide a coordination framework able to support the previous different coordination issues in coherent way, with an engineering perspective aiming at producing reusable and generic components, and in accordance with existing software standards".

Regarding related literature, we can notice that autonomic coordination in dynamic IOW has been insufficiently examined, and tools developed for static IOW cannot be directly adapted notably because they do not take into account three main features of dynamic IOW: autonomy preservation, openness and scalability. Consequently, IOW coordination should be revisited and adapted in this highly dynamic context.

However, we can find some interesting propositions dealing with coordination in dynamic IOW (Buhler and Vidal, 2004; Blake and Gomaa, 2005; Savarimuthu *et al.*, 2005).

Unfortunately, these propositions only provide coordination architecture by defining mediator components (brokers) without specifying the interaction rules or protocols between the different elements of the architecture. Moreover, they do not explain how the IOW process manages unexpected situations like for instance dynamic choice of a partner, its unavailability.

The contribution of the paper is a model for autonomic coordination of business processes in dynamic IOWs. This model relies on the combined exploitation of agent and semantic web approaches viewed, as defended in Section 2, as enabling technologies to deal with the computing context in which IOW is enacted nowadays.

This agent-based coordination model aims at managing the interactions between the different involved business processes while ensuring the preservation of each business process autonomy, and the coherence and effectiveness of the global system, i.e., the IOW. Following the format usually applied for defining agent-based coordination models (Busi *et al.*, 2001), our proposition is a triplet (E, M, R) corresponding respectively to (i) the set of coordinated entities, (ii) the media or infrastructure supporting the coordination and iii) the set of rules governing the coordination.

In our context, E corresponds to business process, called Business Process (BP) services (i.e., services implementing BPs), that may be published, discovered or deployed by IOW partners. BP services are formalized using Petri Nets with Objects (PNO), published in the OWL-S format, and can be played by business processes involved in IOW. M is a multi-agent architecture compliant with the Workflow Management Coalition architecture (Workflow Management Coalition, 1994) and integrating specific components devoted to autonomic coordination, notably mediators for finding IOW partners. Finally, R is described through an organizational model compliant with the Agent Group Role meta-model of Ferber (Ferber *et al.*, 2003) and aiming at ruling and structuring the interaction among the coordinated BP services and the different components of the architecture. It dynamically allocates roles to each partner intervening in an IOW and organizes their interactions in coherent units, termed groups, according to the interaction context (negotiation, finding partners). Agents communicate with FIPA-ACL but the message content language is variable.

The paper is organized as follows. Section 2 highlights the interest of an agent and semantic web-based approach to deal with autonomic coordination of business processes involved in dynamic IOW. Sections 3, 4 and 5 describe the (E, M, R) model we propose for this autonomic coordination. Section 3 presents our semantic web approach to describe BP services that define the entities of our model. Section 4 introduces the agent-based architecture that defines the media of this model while Section 5 explains how we rule the proposed coordination model defining an agent-based organizational model that structures the interaction between the agents of the architecture and the different BP services involved in an IOW. Because of space limitation, the paper only introduces our proposition to deal with the first coordination problem presented before, i.e., finding partners. Finally, Section 6 compares our proposition according to related works and then concludes the paper giving some directions for future works.

2. Why an Agent and Semantic Web-Based Approach?

In our opinion, the combination of these two technologies is suitable for coordination of business processes involved in dynamic IOW.

As introduced before, dynamic IOW is a specific case of IOW where partners are not necessarily known before or during the IOW execution either because they are unknown at design-time or no more available at run-time. Consequently it is necessary that the different Workflow Management Systems (WfMS) involved in a dynamic IOW support specific tools and interfaces to face this highly dynamic context: a dynamic IOW, where (re-)action is necessary to dynamically choose a partner or to face a partner failure, cannot be monitored as a static IOW.

Dynamic coordination was investigated in multi-agent systems area and specific mediators were introduced to deal with agent finding, negotiation or contract specification between agents (Jennings *et al.*, 2001; Wooldridge, 2002; Klusch *et al.*, 2006). So, having an agent view of the coordination issue in dynamic IOW, we could inherit from numerous concrete solutions proposed in this area to deal with agent coordination (Heiberg *et al.*, 2002).

Moreover, agent technology provides natural abstractions to design and model dynamic IOW taking into account the specific features of this context, i.e., distribution, heterogeneity, autonomy and openness:

- Regarding autonomy, agent technology permits to model each local business process as an entity (agent) representing an organization responsible for a mission that it is able to cooperate with other entities (i.e., enterprises/organizations involved in an IOW). Consequently, this technology provides conceptual simplicity and enables us to provide each entity with specific mechanisms in order to be able to decide by itself when, how and why a business process must be provided.
- Distribution is a fundamental concept in Multi-Agent Systems (MAS). Indeed, in MAS, the system is viewed as set of specialized, autonomous and cooperative entities. To support such a view, and thus to support distribution and reduce the complexity of the system design, agent technology introduces specific architectures (Blackboard, BDI), languages (KQML, Fipa-ACL) and protocols (Contract-net, Auction).
- Heterogeneity is supported through agent communication languages of MAS, which ease the communication between heterogeneous agents since they permit to refer an ontology and consider the language used for the message specification as a parameter.
- Finally, MAS permits the description of the organizational dimension of a system to structure and rule both the arrival/departure of agents and the communication between them, using concepts such as role, capacity, group, team, interactions. These concepts are very useful to solve security, scalability and openness problems that may occur in the context of dynamic IOW (Ferber *et al.*, 2003).

On the other hand, the semantic web constitutes a complementary enabling technology (Podgorelec *et al.*, 2007). It first helps to represent a shared business view, through a common terminology or an ontology, without which it would not be possible to solve the various semantic conflicts that are bound to occur between heterogeneous and autonomous business processes involved in a dynamic IOW. Moreover, in open and dynamic environments, where IOW partners are numerous and not known a priori, the semantic web also provides means to describe, publish and discover business processes, called business process services, offered by involved partners.

The three following sections present the agent-based model introduced to deal with autonomic coordination in dynamic IOW. This model is described as a triplet (E, M, R) and these three sections respectively present each element of the triplet.

3. E: Business Process Services

The different business processes involved in an IOW define the Entities to be coordinate. In this section, we defend the idea that the combined use of the Petri Nets with Objects (PNO) formalism and OWL-S semantic web language is convenient for business process specification, validation and publication (Andonoff *et al.*, 2005). In this paper, the different published business processes are called Business Processes (BP) services.

This section briefly presents the two formalisms namely PNO (Sibertin-Blanc, 1985) and OWL-S. It then shows how we combine the use of these two formalisms for BP service specification, validation and publication.

3.1. *OWL-S: A Convenient Language for BP Services Publication*

OWL-S is a semantic markup language that enables the description of web services in order to be selected, invoked and composed (OWLSC, 2005). OWL-S refers to an ontology of services that defines and structures the concepts for handling Web services. The resulting conceptual model is defined through a hierarchy of classes that may be variably refined according to the business domain considered. The essential properties of a service are described by the three following classes: ServiceProfile, ServiceModel and ServiceGrounding.

The ServiceProfile class describes the service in terms of attributes that will be used to find it or select it. These attributes respectively define the required inputs of the service, the outputs it is able to produce, the constraints that must be satisfied by the inputs, and the output properties guaranteed after its execution. The ServiceModel class describes the service in terms of a process model composed of two specifications: a service process and a process control. The ServiceProcess defines the structure of the process using three types of processes: atomic, simple and composite processes. Atomic processes correspond to operations that the service can directly execute; they have no sub-processes. Simple processes correspond to abstractions of atomic processes and are not directly invocable. Composite processes are collections of processes coordinated by control constructs including sequence, loops, conditionals and concurrency. Four attributes are defined for these processes: inputs, outputs, preconditions and effects having the same semantic as the functional capabilities of the ServiceProfile. Regarding the ProcessControl, OWL-S informally represents all the useful attributes for monitoring the execution of the service, notably its possible states at run-time (e.g., ready, ongoing, suspended, aborted). Finally, the ServiceGrounding class defines how to access to the service by specifying the communication protocols and messages, and the port numbers to be used.

OWL-S is appropriate to BP service publication for two main reasons. The first reason is that OWL-S has an adequate expressive power to describe business processes. Indeed, it is possible to describe, using the service profile and the service model of OWL-S, the three different interrelated models of a workflow/business process, i.e., the organizational, informational and process models (van der Aalst, 1998). Regarding the process, there is a direct mapping between the service model of OWL-S and the process model of a workflow. In OWL-S, the described service is broken down into tasks and their coordination is specified using control constructs such as sequence, loops, conditional, concurrency. In the process model of a workflow, the process is also broken down into tasks and workflow patterns are used to coordinate them. Even if OWL-S does not include all the PNO patterns, it provides the control constructs necessary to describe the majority of workflow process models since it allows the modeling of sequence, loops, conditional and concurrency. Regarding the informational and organizational aspects, OWL-S, through the service profile and the service process classes, gives a support for the description of actors, information (data or documents) and their availability as required in a workflow. This is possible thanks to the set of inputs (actors, data and documents), preconditions (actors able to play specific roles, empty documents), outputs (data and documents) and effects (documents well filled, compliant with a specific norm).

Moreover, OWL-S, which is a semantic web service language, enriches web services description based on WSDL with semantic information about the properties (ServiceProfile) and the structure of the service (ServiceModel). Moreover, this semantic information is based on an ontology, extensible according to the domain and described with a well defined markup language. Ontology makes possible, in the context of IOW in which several heterogeneous organizations cooperate, to solve semantic conflicts between these organizations by defining a shared business view based on a common vocabulary. Moreover, OWL-S ontology has a first-order logic representation that permits deduction and eases the implementation of matchmaking mechanisms useful to compare business process services. Such mechanisms are very important when selecting partners.

But OWL-S has an important drawback. It does not provide with theoretical foundations giving an operational semantics to the behavior of the modeled services: thus it is impossible to analyze, simulate and validate the modeled BP services. Now, in our opinion, it is important, indeed fundamental, to stand guarantees for the behavior of the offered services. Consequently, to compensate this drawback, we propose the use of a graphical and formal language, namely Petri Nets with Objects (PNO), to specify and validate BP services. PNO (Sibertin-Blanc, 1985) are used as a graphical tool to help a designer to define a BP service; they also provide formal and executable specifications to analyze, simulate, check and validate the described BP service behavior.

3.2. *PNO: A Convenient Language for BP Services Specification and Validation*

Petri Nets with Objects (PNO) (Sibertin-Blanc, 1985) are a formalism combining coherently Petri nets (PN) technology and the Object-Oriented (OO) approach. While PN are very suitable to express the dynamic behavior of a system, the OO approach permits the modeling and the structuring of its active (actor) and passive (information) entities. In a conventional PN, tokens are atomic, whereas they are objects in a PNO. As any PN, a PNO is made up of places, arcs and transitions, but in a PNO, they are labeled with inscriptions referring to the handled objects.

Petri nets are widely used for workflow service specification (van der Aalst, 1998). Several good reasons justify their use:

- An appropriate expressive power that permits the description of the different tasks involved in a business process and their coordination.
- An operational semantics making an easy mapping from specification to implementation possible.
- Theoretical foundations permitting analysis and validation of behavioral properties and simulation facilities.

Unfortunately, conventional Petri nets focus on the process definition and do not perfectly capture the organizational and the informational dimensions of BPs. Petri nets with Objects extend Petri nets by integrating high-level data structure represented as objects, and, therefore provide the possibility to integrate the two dimensions missing in Petri nets. Thus, using PNO, actors of the organizational model are directly represented as objects and they may be invoked through methods in the action part of a transition. In the same way, data and documents of the informational model are also represented by objects flowing in the PNOs and transformed by transitions. To summarize, PNOs are convenient for business process specification because they really are a strong link between the three workflow models since they permit the description, in a same representation, of actors of the organizational model, data and documents of the informational model, and tasks and their coordination of the process model. Moreover, we use PNO as a graphical tool to specify a BP service and as a formal tool to define executable specifications in order to analyze, simulate, check and validate a BP service. So, PNO makes OWL-S complete providing it with a graphic representation and an operational semantics of the specified BP services.

3.3. *Combining PNO and OWL-S for Business Process Service Specification, Validation and Publication*

Consequently, we use both PNO and OWL-S for BP service specification, validation and publication. As illustrated in Fig. 1, the modeling process of a business process service includes a graphical specification and validation step. Then, after being sure to guarantee the quality of the specified BP service, we automatically derive the corresponding OWL-S specifications in order to publish it.

This specifying process and more particularly, the automatic derivation of OWL-S specifications from PNO ones, is not presented in the paper. The interested reader can

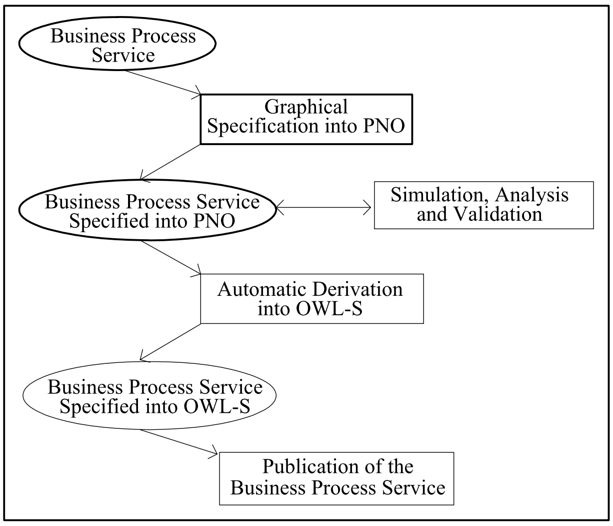

Fig. 1. Specifying process of a business process service.

consult (Andonoff *et al.*, 2005) for further information (tables for the mapping between PNO and OWL-S concepts and algorithms implementing this mapping).

4. Media (M): An Agent-Based Architecture for Autonomic Coordination in Dynamic IOW

This section presents the agent-based architecture that defines the Media (M) of our coordination model. This architecture provides a solution to deal with IOW coordination issues but, because of space limitation, this paper only introduced the solution we provide to deal with partners finding.

The reference architecture of a WfMS, defined by the Workflow Management Coalition (Workflow Management Coalition, 1994), is at the basis of our proposition. This architecture supports syntactic interoperability between WfMSs and communication between workflows. Thus, as defended in Buhler and Vidal (2005), it is highly appropriate to be conformed with this architecture in the context of IOW. According to this architecture, a WfMS includes a Workflow Enactment Service (WES) and supports the following interfaces:

- Interface 1 with Process Definition,
- Interface 2 with Workflow Client Applications,
- Interface 3 with Invoked Applications,
- Interface 4 with others WESs,
- Interface 5 with Administration and Monitoring.

In the IOW context, the two main components of this architecture are the WES and Interface 4. The aim of the WES, on which one imposes no constraints upon its internal structure, is to manage the execution of one or several instances of business processes while the aim of Interface 4 is to connect WfMSs together in order to share the execution of a workflow process between different WESs of different organizations.

However, this reference architecture is inadequate in the context of dynamic IOW since the WES must not only execute business processes but also must mix different concurrent activities including finding partners, negotiation between partners, contract specification and cooperation with other WESs offering and providing BP services.

4.1. *Revisiting the WES with Agents*

Fig. 2 explains how we have rethought the Workflow Enactment Service using agents. This architecture includes (i) as many Workflow Agents as running business process instances, (ii) an Agent Manager responsible for these Workflow Agents, (iii) a Connection Server that interacts with mediation infrastructures specialized in partners finding, negotiation between partners, contract specification, and, (iv) a new interface, Interface 6, to support the communication between a connection server and these mediation infrastructures.

Regarding Workflow Agents, the idea is to implement each business process instance (stored in the Business Processes database) as a software process, and to encapsulate this instance within an (pro-)active agent. Such a Workflow Agent includes a workflow engine that, as and when the business process instance progresses, reads the BP schema definition (specified in OWL-S), and triggers the action(s) to be done according to its current state. This Workflow Agent supports interface 3 with the applications that are to be used to perform pieces of work associated to process' tasks.

The Agent Manager controls and monitors the running of Workflow Agents:

- Upon a request for a new instance of a business process, the Agent Manager creates a new instance of the corresponding business process agent schema, initializes its parameters according to the context, and launches the running of its workflow engine.

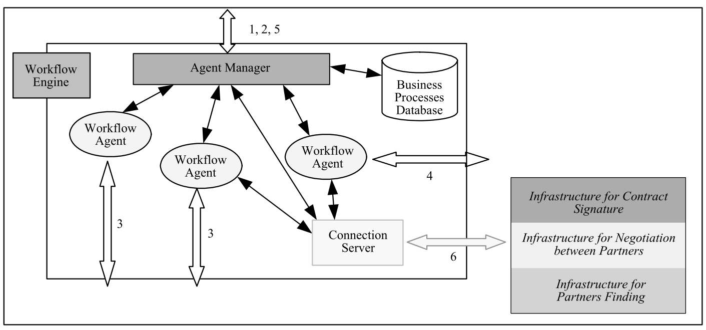

Fig. 2. Agent architecture for the workflow engine.

- It ensures the persistency of Workflow Agents that execute long-term business processes extending to a long time in which task performances are interleaved with periods of inactivity.
- It coordinates Workflow Agents in their use of shared resources.
- It assumes interfaces 1, 2 and 5 of the WfMS with the environment.

In the IOW context, Workflow Agents belonging to an organization (called requester Workflow Agents) need to find other agents running in other organizations (called provider Workflow Agents) and able to contribute to the achievement of their goal. However, the capacity, the availability or even the presence of such provider Workflow Agents is not steady. Finding them or negotiate with them requires maintaining knowledge about resources in the environment.

The role of the Connection Server is to store this knowledge in a specific database (not visualized in Fig. 2) and to help requester Workflow Agents to find the partners they need. More precisely, the Connection Server interacts with mediation infrastructures specialized in finding partners, negotiation between partners, contract specification. This requires to define a new interface, Interface 6, that supports the communication between a connection server and the mediation infrastructures, and, consequently, between different connection servers of different Workflow Enactment Services. Interface 4 of the reference architecture cannot be used for such a communication since it only supports the execution of a business process between different workflow engines.

In our proposition, we consider coordination as a specific component when designing and implementing an IOW, and consequently, we separate coordination activities from business processes execution. That is why we introduced mediation infrastructures dedicated to partners finding, negotiation between partners and contract specification. These infrastructures are of course independent of the WES, and are dynamically plugged to it only when it is necessary. In fact, we have applied the principle of separation of concerns, which allows the separation of individual and intrinsic capabilities of each workflow system from what is related to IOW coordination. This principle is widely recognized as a good design practice from a software engineering point of view (Ghezzi *et al.*, 1991) and has led to the advent of new technologies in Information Systems as discussed in (van der Aalst, 1998).

Finally, the internal architecture of a Workflow Agent permit it to face unexpected situations like for instance dynamic choice of a partner (in order to take into account the context for the best), unavailability of a partner (a partner is no more able to provide the requester business process service). This internal architecture is presented in Fig. 3.

A Workflow Agent is implemented as three functional agents: (i) a Case Agent, which is responsible for the execution of the Workflow Agent's business process, (ii) an Introspector Agent, which checks that every external execution of a Case Agent is possible, and, (iii) a Coordination Process Agent which is, if necessary, launched by the Introspector Agent in order to help it for finding partners, negotiating or signing contracts with them. There are as many coordination process agents for a case agent as the number of impossible external execution supported by an IOW partner.

Case Agents and Coordination Process Agents each include a workflow engine: the workflow engine of a Case Agent executes the business process (described in OWL-S)

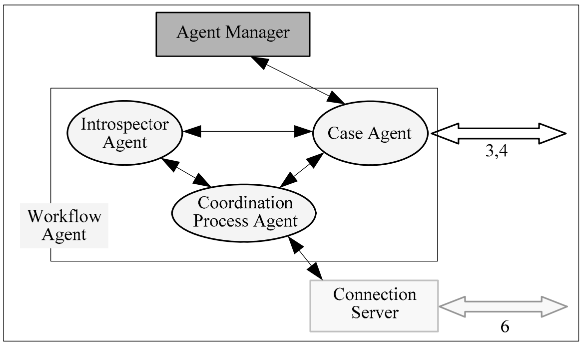

Fig. 3. Internal architecture of Workflow Agents.

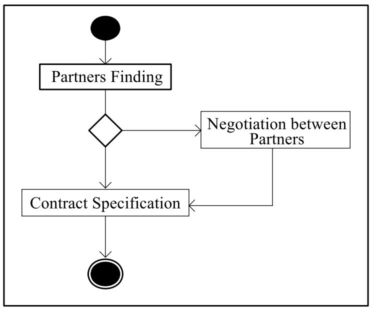

Fig. 4. UML description of the coordination process.

it represents while a Coordination Process Agent executes a process which implements all the coordination activities: partners finding, negotiation between partners and contract signature between partners, as illustrated in Fig. 4. Consequently, Case Agents assume interfaces 3 and 4 while Coordination Process Agents are in connection, via the connexion server, with interface 6.

4.2. *Mediation Infrastructure for Partners Finding*

We have introduced a mediation infrastructure specialized in partners finding. This infrastructure is dynamically plugged to the WES using the Connection Server and Interface 6. This infrastructure is given in Fig. 5. It includes (i) a Matchmaker whose aim is to connect requester Workflow Agents to provider Workflow Agents, (ii) an Ontology database used to solve semantic interoperability problems that obviously occur between requester and provider Workflow Agents, and (iii) a Business Process Services database that stores offered business process services.

The aim of this infrastructure is to connect requester Workflow Agents looking for Business Process (BP) services to provider Workflow Agents able to implement these BP services according to the following protocol: (i) an Agent Manager belonging to a provider organization advertises the proposed BP service to the Matchmaker,

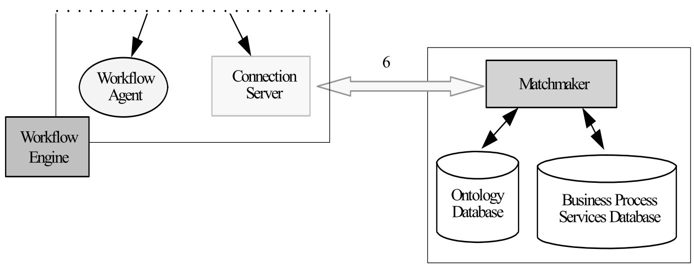

Fig. 5. Mediation infrastructure for partners finding.

(b) the Matchmaker stores the service in its Business Process Services database, (c) a requester Workflow Agent belonging to a requester organization looking for a BP service asks the Matchmaker whether it knows provider organizations offering the desired service, and finally (d) the Matchmaker matches the request against the stored services and returns the result as a set of BP service providers (more precisely a set of connection servers representing the provider organizations).

The Matchmaker compares offers and requests of BP services and returns the identity of the providers able to realize the requested BP service. Data semantic interoperability problem between offers and requests is solved using an ontology of the Ontology database. On the one hand, the ontology defines the offer and request BP service vocabulary and, on the other hand, the ontology permits, as illustrated in Bouzguenda *et al.* (2008), intelligent comparisons between offers and requests including subsumption (Klusch *et al.*, 2006).

5. Rules (R): An Agent-Based Organizational Model for Dynamic IOW

This section presents the architecture that defines the Rules of our coordination model. We have adopted an organizational view providing macro-level coordination rules to specify and describe how IOW agents, corresponding to both mediation infrastructures' agents and WfMS' agents (BP service agents) interact among themselves. The organizational model structures the communication between the IOW agents and thus highlights the coordination of the different organizations involved in an IOW during finding partners, negotiation between partners, signature of contracts.

For that purpose, we use the Agent Group Role (AGR) meta-model (Ferber *et al.*, 2003) which is a possible framework to define organizational dimension of a MAS and which is particularly appropriate for dynamic IOW (Artikis, 2001).

The remainder of this section first describes AGR and then shows how, using this meta-model, we structure and rule the interactions between the IOW agents: because of space limitation, we only present, as an example, how we rule the interactions between agents during the partners finding step. Finally, this section gives an AUML Sequence Diagram that illustrates the exchange of messages between these agents.

5.1. *Organizational Modeling with AGR*

We have adopted an organizational view that provides appropriate concepts to describe the macro-level of the coordination among partners in terms of externally observable behavior, independent of the internal features of each participating component. These concepts (such as roles, groups, teams, interactions or commitment) are conceptual tools that ease the modeling of the structure and behavior of the systems under consideration (Chen *et al.*, 2008).

In our context, these concepts allows to describe, at a macro-level, the IOW' agents coordination when partners finding, negotiation between partners and contract specification, and consequently, to highlight the coordination between organizations involved in a dynamic IOW. These concepts also provide means to deal with security, scalability and openness problems.

The Agent, Group and Role (AGR) meta-model, first introduced in Ferber and Gutknecht (1998) and then extended in Ferber *et al.* (2003), is one of the frameworks proposed to define the organizational dimension of a multi-agent system, and it is well appropriate to the IOW context (Andonoff *et al.*, 2006). According to this meta-model, the organization of a system is defined as a set of related groups, agents and roles. A group is a set of agents that also determines an interaction space: an agent may communicate with another agent only if they belong to the same group. The cohesion of the whole system is maintained by the fact that agents may belong to any number of groups, so that the communication between two groups may be done by agents that belong to both. Each group also defines a set of roles, and the group manager is the specific role fulfilled by the agent that initiates the group. The membership of an agent in a group requires that the group manager authorizes this agent to play a role, and each role determines how the agents playing it may interact with other agents. So the whole behavior of the system is framed by the structure of the groups that may be created and by the behaviors given to the agents by the roles. On the other hand, the AGR meta-model agrees with any kind of agents since it imposes no constraint on their structure.

5.2. *An Organizational Model for Partners Finding*

The organizational model we propose to describe and structure the coordination between IOW' agents when partners finding is presented in Fig. 6. It is organized around the following components:

- Four groups represented by ellipses which are: BPServiceOffer, BPServiceRequest, Offer and Request.
- Five agents represented by candles which are: ProviderConnectionServer, RequesterConnectionServer, RequesterWorkflowAgent, AgentManager and Matchmaker.
- Eight roles, represented par cardinalities (1 or *), since each agent plays a specific role within each group.

Let us detail now how each group operates. On the one hand, the BPServiceOffer group enables one or several (the multiplicity is represented by a star inside the role)

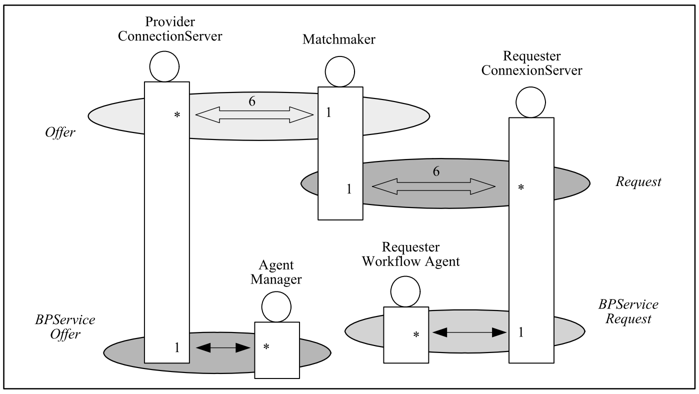

Fig. 6. Organizational model to support partners finding.

agent managers (represented by the AgentManager agent) to enter in connection with their connection server (represented by the ProviderConnectionServer agent) to publish an offer of a Workflow Service. The ProviderConnectionServer agent forwards this offer to the Matchmaker agent (Offer group).

On the other hand, the BPServiceRequest group enables one or several requester workflow agents (represented by the RequesterWorkflowAgent agent) to enter in connection with their connection server (represented by the RequesterConnectionServer agent) to which the search of a partner (a provider organization) will be delegated. The RequesterConnectionServer agent then forwards this request to the Matchmaker agent (Request group). The Matchmaker agent matches the offers stored in the Business Process Service database and returns to the RequesterWorkflowAgent agent the different connection servers of organizations able to provide the requested BP service.

5.3. *Exchange of Messages between Agents*

The standard FIPA-ACL communication language (FIPA, 2002) is used to support the interaction, through messages exchange, between the different agents involved in the organizational model. FIPA-ACL offers a convenient set of performatives to deal with the different coordination issues introduced before (for instance, agree, cancel, refuse, request, inform, confirm for finding partners, or propose, accept-proposal for negotiation between partners). Moreover, FIPA-ACL supports messages exchange between heterogeneous agents since (i) the language used to specify the message is free and (ii) a message can refer to an ontology. This latter point is very interesting since it is possible, through FIPA-ACL messages, to refer a domain ontology, which can be used to solve semantic interoperability problems (Andonoff *et al.*, 2006).

Fig. 7 illustrates, giving an AUML Sequence Diagram, this messages exchange during the partners finding step. This sequence diagram shows the FIPA-ACL interactions between agents belonging to the groups previously introduced in Fig. 6 which are BPServiceOffer, BPServiceRequest, Offer and Request groups.

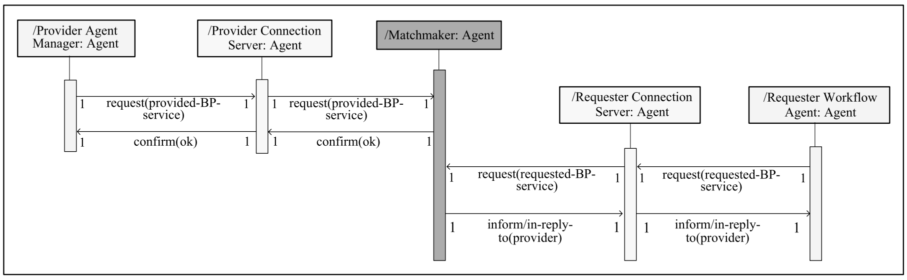

Fig. 7. AUML sequence diagram illustrating messages exchange.

6. Discussion and Conclusion

This paper has addressed autonomic coordination issue in dynamic Inter-Organizational Workflow (IOW). As indicated in the introduction, giving the maturity of Internet standards (SOAP, WSDL, UDDI and BPEL for instance) for enterprise cooperation, dynamic IOW is more and more required and autonomic coordination of business processes involved in such IOW is now a topical issue.

This paper has tried to answer to the following questions: how to automatically and autonomously maintain the coordination of different business processes involved in an IOW? How to face the unavailability at run-time or the not knowledge at design-time of an organization responsible for the execution of a business process involved in an IOW?

To reach this objective, this paper has:

- Defended the idea that the combined used of agent technology and semantic web is well suited to deal with autonomic coordination in dynamic IOW.
- Specified a multi-agent based model which is a triplet (E, M, R) corresponding respectively to (i) E: the set of coordinated entities, (ii) M: the media or infrastructure supporting the coordination and (iii) R: the set of rules governing the coordination. This model has been implemented in order to validate the feasibility of our approach. Because of space limitation, this paper does not report this implementation. The interested reader can consult Bouzguenda (2006) for details on it and its evaluation.

This work has two originalities. The first one lies on the combination of agent and web semantic technologies for modelling IOW. Well articulated, these two technologies bring adequate coordination mechanisms for coordination in IOW. On the one hand, modelling IOW using agents permits to bring business process coordination issue back onto agent coordination issue. As this issue has been deeply investigated in multi-agent systems area, it is possible to benefit from the results obtained in this area to deal with autonomic coordination of business processes involved in dynamic IOW. On the other hand, the semantic web helps to represent a shared business view, through a common terminology or an ontology, without which it would not be possible to solve the various semantic conflicts that are bound to occur between several heterogeneous and autonomous business processes involved in a dynamic IOW. Moreover, in open and dynamic environments, where IOW partners are numerous and not known a priori, the semantic web also provides means to describe, publish and discover business processes, offered by involved partners.

The second originality is to investigate coordination as a whole dimension when designing and implementing dynamic IOW. Instead of considering coordination in a distributed manner, i.e., dissolved in each business process involved in IOW, on the contrary, we have isolated it, specified it and entrusted it to mediator components shareable by participating business processes (Hanachi and Sibertin-Blanc, 2004; Andonoff *et al.*, 2007). Hence, from en engineering point of view, the coordination design, validation, development and maintenance are some facilitated.

Regarding related works, IOW coordination already gave place to several works and systems such as CrossFlow (Grefen *et al.*, 2000), Derpa (Casati and Discenza, 2000), or Yawl (van der Aalst *et al.*, 2004) for instance. These systems mainly address coordination issue in static IOW: they assume to know a priori the partners involved in the IOW and therefore do not provide means for dynamically finding unknown partners, negotiating or contracting with them during the IOW process execution.

Agent technology was also used to study and implement workflow applications. Among these works we can especially quote those of (Singh and Huhns, 1999; Yan *et al.*, 2001; Savarimuthu *et al.*, 2005; Zeng *et al.*, 2001), which adopt an approach similar to our by entirely rethinking the design and the development of WfMS in terms of agents. The first three works use agent technology to make processes more flexible and adaptive but they do not address the cooperation between several processes or workflow systems. Zeng *et al.* (2001) addresses partners finding, but neither deals with negotiation between partners or contract specification issues, nor meets the requirements of the WfMC coalition. Finally, it does not exploit the advantages of the semantic web to deal with semantic heterogeneity issues encountered in dynamic IOW.

The idea of combining Agent and Web services in workflow also exists in the works of Buhler and Vidal (2005), Aberg *et al.* (2005) and Blake and Gomaa (2005). Buhler and Vidal (2005) defends the idea that multi-agent systems and Web services can be mixed for flexible enactment of enterprise workflow: agents are responsible for coordination while Web services correspond to pieces of code to be coordinated. More precisely, Buhler and Vidal (2005) proposes the use of BPEL for web services as specification language for expressing the initial social order of the multi-agent system. Unfortunately, this work neither give details about the architecture which underlies their proposition, nor about the means used to support unexpected situations at run-time like partner failure, dynamic choice of a partner Aberg *et al.* (2005) proposes an agent-based architecture compliant with the Workflow Management Coalition architecture and supporting the search of web services composing workflow processes. This work proposes a solution to coordinate a centralized dynamic IOW which supports the search of services. Consequently, it differs from our since we provide a solution for distributed execution of dynamic IOW which correspond to a more general case of dynamic IOW. Finally, Blake and Gomaa (2005) defines the WARP architecture for cross-organizational workflow. This architecture mixes agents and Web services. Web services correspond to applications to be orchestrated and agents are responsible for the composition and orchestration of Web services since the role of the agents is to make evolution of service-based workflow composition possible. The WARP architecture also supports brokering for partners finding. Although very complete and interesting, this work only support centralized IOW execution.

To sum up, these works have in common the proposition of a solution for centralized execution of dynamic IOW. Each task involved in such an IOW is described as a Web service and the agent-based approach is used to add flexibility to the IOW supporting brokering of services.

These works differ from our for several reasons. First, we provide a solution for distributed execution of dynamic IOW. Second, we highlight how our workflow agents face unexpected situations at run-time. Third, coordination issues are not limited to partners finding as we also consider negotiation between partners (Andonoff and Bouzguenda, 2005), and contract signature (Combettes *et al.*, 2006). Fourth, following a semantic web-based approach, we propose a solution which:

- solves semantic heterogeneity problems that obviously occur in dynamic IOW;
- improves the comparison mechanism between requested and offered business process services during the partners finding step;
- really takes into account the three dimensions of a business process, i.e., the informational, organizational and process dimensions.

Regarding current works, our effort focuses on the implementation of the proposed model. We have already validated the feasibility of our solution developing a simulator, called Matchflow, which implements a part of the model (Bouzguenda, 2006). More precisely, Matchflow implements the publication process of OWL-S business processes (E), some agents of the proposed architecture (M), and also the corresponding organizational model (R). For this development, we have used the Madkit multi-agent platform (Ferber and Gutknecht, 2004), which integrates the Agent Group Role meta-model permitting the description of the organizational dimension of a multi-agent system. Matchflow mainly focus on the Matchmaker agent. We have implemented different comparison modes (Subsume, Plug-In and Exact) and have tested the Matchmaker efficiency designing and implementing a specific dynamic IOW case study: the "reviewing papers" case study (Bouzguenda, 2006).

Currently, our objective is to develop a solid prototype that will implement all the agents presented in the proposed architecture, notably the workflow agents. To reach this objective we plan to use the WADE multi-agent platform (Bellifemine *et al.*, 2007) since this development environment facilitates the implementation of agents integrating workflow engines.

References

van der Aalst, W. (1998). The application of Petri-nets to workflow management. *Circuit, Systems and Computers*, 8(1), 21–66.

van der Aalst, W. (1999). Inter-organizational workflows: An approach based on message sequence charts and Petri nets. *Systems Analysis, Modeling and Simulation*, 34(3), 335–367.

van der Aalst, W., Aldred, L., Dumas, M., ter Hofstede A. (2004) Design and implementation of the YAWL system. In: *Int. Conference on Advanced Information Systems Engineering*, Riga, Latvia, pp. 142–159.

Aberg, C., Lambrix, C., Shahmehri, N. (2005). An agent-based framework for integrating workflows and web services. In: *Int. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Linköping, Sweden, pp. 27–32.

Andonoff, E., Bouzguenda, L. (2005). Agent-based negotiation between partners in loose inter-organizational workflow. In: *Int. Conference on Intelligent Agent Technology*, Compiègne, France, pp. 619–625.

Andonoff, E., Bouzguenda, L., Hanachi, C. (2005). Specifying workflow web services for finding partners in the context of loose inter-organizational workflow. In: *Int. Conference on Business Process Management*, pp. 120–136.

Andonoff, E., Bouzguenda, L., Hanachi, C. (2006). La coordination dans le workflow inter organisationnel lâche : une approche basée sur les agents et le web sémantique. *Ingénierie des Systèmes d'Information*, 11(4), 127–150.

Andonoff, E., Bouaziz, W., Hanachi, C. (2007). A protocol ontology for inter-organizational workflow coordination. In: *Int. Conference on Advances in Databases and Information Systems*, Varna, Bulgaria, pp. 28–40.

Artikis, A., Pitt, J. (2001). A formal model of open agent societies. In: *Int. Conference on Autonomous Agents*, Montreal, Canada, pp. 192–193.

Bellifemine, F., Caire, G., Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Wiley.

Blake, B., Gomaa, H. (2005). Agent-oriented compositional approaches to services-based cross organizational workflow. *Decision Support Systems*, 40(1), 31–50.

Bouzguenda, L. (2006). *Coordination Multi-Agents pour le Workflow Inter Organisationnel Lâche*. Phd Dissertation, laboratoire IRIT, Université de Toulouse 1.

Bouzguenda, L., Bouaziz, R., Andonoff, E. (2008). Using ontologies for coordination in loose inter-organizational workflow. In: *Int. Conference on Research Challenge in Information Science*, Marrakech, Morocco, pp. 137–146.

Buhler, P., Vidal, J. (2005). Towards adaptive workflow enactment using multi-agent systems. *Information Technology and Management*, 6(1), 61–87.

Busi, N., Ciancarini, P, Gorrieri, R., Zavattaro, G. (2001). Coordination models: A guided tour. In: Omicini, A., Zambonelli, F., Klusch, M., Tolksdorf, R. (Eds,.), *Coordination of Internet Agents*, Springer-Verlarg. pp. 6–24.

Casati, F., Discenza, A. (2000). Supporting workflow cooperation within and across organizations. In: *Int. Symposium on Applied Computing, Track on Coordination Models, Languages and Applications*, Como, Italy, pp. 196–202.

Chebbi, I., Dustdar, S., Tata, S. (2006). The view based-approach to dynamic inter-organizational workflow cooperation. *Data and Knowledge Engineering*, 56(2), 139–173.

Chen, N., Yu, Y., Ren, S., Bechman, M. (2008). A role-based coordination model and its realization. *Informatica*, 32(3), 229–244.

Combettes, S., Hanachi, C., Sibertin-Blanc, C. (2006). Organizational Petri nets for protocol design and enactment. In: *Int. Conference on Autonomous Agents and Multi-Agents Systems*, Hokodate, Japan, pp. 1384–1386.

Divitini, M. Hanachi, C., Sibertin-Blanc, C. (2001). Inter-organizational workflows for enterprise coordination. In: Omicini, A., Zambonelli, F., Klusch, M., Tolksdorf, R. (Eds.), *Coordination of Internet Agents*, Springer-Verlarg, pp. 46–77.

Ferber, J., Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. In: *Int. Conference on Multi-Agents Systems*, Paris, France, pp. 128–135.

Ferber, J., Gutknecht, O. *The MadKit Project: A Multi-Agent Development Kit*. Available at: `http://www.madkit.org`.

Ferber, J., Gutknecht, O., Fabien, M. (2003). From agents to organizations: An organizational view of multi-agent systems. In: *Int. Workshop on Agent-Oriented Software Engineering*, Melbourne, Australia, pp. 214–230.

Foundation for Intelligent Physical Agents (2002). FIPA ACL. *Message Structure Specification*. Available at: `http://www.fipa.org/specs/fipa00061/`.

Ghezzi, C., Jazayeri, M., Mandrioli, D. (1991). *Fundamentals of Software Engineering*. Prentice-Hall.

Grefen, P., Aberer, K., Hoffner, Y., Ludwig, H. (2001). CrossFlow: Cross-organizational workflow management in dynamic virtual enterprises. *Computer Systems Science and Engineering*, 15(5), 277–290.

Hanachi, C., Sibertin-Blanc, C. (2004). Protocol moderators as active middle-agents in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 8(3), 131–164.

Heiberg, T., Matskin, M., Pederseb, J. (2002). An agent-based architecture for customer services management and product search. *Informatica*, 13(4), 441–454.

Jenning, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., Wooldridge, M. (2001) Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10(2), 199–215.

Klusch, M., Fries, B., Sycara, K. (2006). Automated semantic web service discovery with OWLS-MX. In: *Int. Conference on Autonomous Agents and Multi-Agents Systems*, Hakodate, Japan, pp. 915–922.

OWL Services Coalition: Ontology Web Language for Services Version 1.0. Documentation available at: `http://xml.coverpages.org/ni2004-01-08-a.html`.

Podgorelec, V., Pavli, L., Hericko, M. (2007). Semantic web-based integration of knowledge resources for supporting collaboration. *Informatica*, 31(1), 85–91.

Savarimuthu, T., Purvis, M., Purvis, MK., Cranefield, S. (2005). Integrating web services with agent based workflow management system. In: *Int. Conference on Web Intelligence*, Compiègne, France, pp. 471–474.

Sibertin-Blanc, C. (1985). High level Petri nets with data structure. In: *Int. Workshop on Petri Nets and Applications*, Espoo, Finland.

Singh, M., Huhns, M. (1999). Multi-agent systems for workflow. *Intelligent Systems in Accounting, Finance and Management*, 8(1), 105–117.

Wooldridge, M. (2002). *An Introduction to Multi-Agent Systems*. Wiley.

Workflow Management Coalition (1994). *The Workflow Reference Model. Technical Report WfMC-TC-1003*.

Yan, Y., Maamar, Z., Shen, W. (2001). Integration of workflow and agent technology for business process management. In: *Int. Conference on Computer Supported Cooperative Work in Design*, Ontario, Canada, pp. 420–426.

Zeng, L., Ngu, A., Benatallah, B., O'Dell, M. (2001). An agent-based approach for supporting cross-enterprise workflows. In: *Australian Database Conference*, Bond, Australia, pp. 123–130.

E. Andonoff received a PhD degree from Paul Sabatier University, Toulouse, France, in 1992. He is currently associate professor at Toulouse University and also member of IRIT (Institut de Recherche en Informatique de Toulouse) Laboratory. His research interest includes inter-organizational workflow, workflow evolution, and protocols in multi-agent systems.

W. Bouaziz is a PhD student at Toulouse University and also member of IRIT (Institut de Recherche en Informatique de Toulouse) Laboratory. His research interest includes multi-agent protocols for coordination in inter-organizational workflow.

C. Hanachi is professor in computer science at University of Toulouse (France) and is a member of IRIT Laboratory (Institut de Recherche en Informatique de Toulouse). His main research interest is coordination in agent oriented information systems. He co-authored more than 50 publications in journals, conferences and book chapters. More information can be found at: `http://w3.univ-tlse1.fr/irit/soc/perso/hanachi/`.

L. Bouzguenda is assistant professor at Higher Institute of Computer Science and Multimedia of Sfax in Tunisia. He is a member of the MIRACL Laboratory. His research works concern coordination and flexibility of cross organizational workflow using agents and web services.

Autonominio tarp organizacinių verslo procesų koordinavimo modelis, grindžiamas agentinėmis technologijomis

Eric ANDONOFF, Wassim BOUAZIZ, Chihab HANACHI, Lotfi BOUZGUENDA

Tarporganizacinės darbų sekos (TDS) apima autonominius ir heterogeninius verslo procesus, kurie yra išsiskirstę atskirose organizacijose ir vyksta joms bendradarbiaujant. Tokių verslo procesų koordinavimas yra viena iš fundamentalių problemų. Tačiau šiuo metu paprastai nagrinėjami tik statikos uždaviniai. Ši problema yra nemažiau aktuali ir sprendžiant dinamikos uždavinius, kurie svarbūs tokiame kontekste kaip internetas, kuriame vis dažniau galime aptikti TDC taikomąją programinę įrangą. Šiai įrangai kurti paprastai taikomos visiems gerai žinomos multiagentinės sistemos. Jos yra žinomos kaip dažniausiai taikomas sprendimas TDC modeliavimui, nes suteikia adekvačias abstrakcijas ir specifinius informacijos nešėjus, padedančius susidoroti su TDC koordinavimo uždaviniais. Straipsnis pristato agentinį modelį, skirtą verslo procesų, dalyvaujančių dinaminėse TDC, koordinavimui. Siūlomą modelį sudaro trys dalys (E, M, R). E yra koordinuojamų esybių aibė. Ji sudaryta iš verslo procesų, kurie gali būti naudojami, atrasti ar įdiegti atskirų TDC partnerių. M yra informacijos nešėjas naudojamas koordinavimui. Jis yra multiagentinės architektūros ir suderinamas su WfMC (angl. Workflow Management Coalition) architektūra bei papildytas specifiniais komponentais specialiai skirtais koordinavimo problemoms spręsti. R yra taisyklių, valdančių koordinavimą rinkinys. R išreiškiamas panaudojant organizacinį modelį, kuriuo siekiama pavaizduoti koordinuojamų esybių ir atskirų architektūros komponentų tarpusavio sąveikų struktūrą.