The NetAcademy ®
www.netacademy.org
NA.editors@netacademy.org

The NetAcademy project has been developed at the
Institute for Media and Communications Management MCM
University of St. Gallen
Blumenbergplatz 9
CH–9000 St. Gallen, Switzerland
www.mcm.unisg.ch

A Conceptual Framework For Agent Oriented and Role Based Workflow Modeling

Lei Yu Beat F. Schmid

Institute for Media and Communications Management
University of St. Gallen
Mueller Friedberg Str. 8
CH-9000 St. Gallen
Switzerland
Email: lei.yu@unisg.ch , beat.schmid@unisg.ch

Abstract. This paper explores an innovative conceptual framework for agent oriented and role based workflow modeling. We view a business process as a collection of autonomous, problem solving agents which interact with others when they have interdependencies. And we model a workflow as a set of relating roles. Roles are defined in term of goals, qualifications, obligations, permissions, protocols, etc. We adopt protocols to govern the interactions among roles. Roles are assigned to agents based on the evaluation of qualification and capabilities. Once a role is assigned to an agent, the agent inherits the obligations and permissions specified in that role. Coordination of workflow is achieved by communication between agents. Moreover, the journal article Internet peer review process is considered as a case study.

1. Introduction

Workflow Management is today considered as one of the key technologies for providing efficiency and effectiveness of organizations. The task for workflow is to describe the coordination and performance of work undertaken in an organization. The current generation workflow management systems usually start modeling workflow from activities analysis. They emphasizes the coordination of activities by highlighting their interdependence. They tend to operate with a central workflow engine which monitors all events in the system. Most of them support planned work. They offer limited support and minimal flexibility during process enactment [15]. In situations where a business process is fully resourced and every conceivable outcome can be considered and controlled, then they should be adequate. However, business environments are now becoming more volatile, and follow more complex processes that require workflow systems, which no longer rely on pre-specified actions. The trend from planned to situated work requires new kinds of workflow support.

One extension of pre-planned workflow is adaptive workflow [12]. The extension is to make execution more adaptive to organizational change. They use an organizational knowledge base to adjust particular tasks to cater for unexpected inputs or changes in the organization. Adaptive workflow are more flexible than pre-planned workflow and usually specify tasks declaratively by describing what has to be achieved in a workflow; it then being left to the workflow management system to deduce the "how" from "what". These are precisely the classes of tasks and environments that Distributed Artificial Intelligence (DAI) research has been investigating in the context of controlling computational entities and physical devices.

With agent technology, a completely new paradigm evolved which promises to tackle the issues of distributed systems more adequately and which has opened up a new dimension of problems that can be addressed. A *Multi-agent Systems (MAS)* can be defined as a loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver. MAS looks at cases where agents have to coordinate themselves in order to achieve their work. In contrast to workflow management, no plan of the work to be done (process or workflow description) is existent before the work starts.

Motivated by such observations, in this paper, we propose an innovative framework for agent oriented and role based workflow management. We view a workflow as a set of autonomous, problem solving agents which interact with others when they have interdependencies. And we model a workflow as a set of relating roles. A role refers to the expected behavior patterns an agent must perform. It has these essential features: goals, qualifications, relationships, obligations, permissions, constraints, protocols, etc. The interactions between agents are governed by speech act based protocols. The coordination of workflow is achieved by the communication of agents.

The remainder of this paper is structured as follows. Section 2 sets up a conceptual framework for agent oriented and role based workflow management. To bridge the gaps between the conceptual architecture and practical applications, in section 3, we propose an approach for developing agent oriented and role based workflow systems. Section 4 is dedicated to applying the approach in modeling the Internet peer review process in detail. In section 5, the related work is briefly reviewed. Finally , in section 6, the work is summarized and prospective future work is also shown.

2. Toward Agent Oriented And Role Based Workflow Management

2.1 Role Based Workflow Modeling

Workflow management systems is a system that defines, creates and manages the execution of workflows through use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and where required, invoke the use of IT tools and applications[24]. Current workflow management systems seem problematic to model processes in a

dynamic social organisation settings, where the business process, unlike automated processes executed by machines, are made up of social actors who have goals and interests, which they pursue through a network of relationships with other actors. A richer workflow model should therefore include not only how activities progress, but also how the actors performing them relate to others intentionally, i.e., in terms of concepts such as actor, goal, dependency, etc. [25].

In a previous work [26], a goal-achieving perspective of workflow has been discussed by using *Malone's coordination theory* [19]. We define a workflow as a coordinated set of interdependent activities which are performed by actors in an organization in order to achieve a set of common goals. *An Activity* is a description of a piece of work that forms one logical step within a process. We further concluded that the most natural way is to view a business process as a collection of autonomous, problem solving agents which interact with others when they have interdependencies[27]. We argue that coordination is achieved by communication between agents in agent-oriented workflow systems. In this paper, an agent is an active entity playing roles within an organization. It can be a human being, a computer system, or an organizational unit. The behavior of an agent is consequence of its mental states such as beliefs, intentions, capabilities, goals, commitments, etc [23].

A workflow instance is a process combined with an organization, and assigns the activities in the process to agents in the organizations. A workflow model is used to create and manage workflow instance in a workflow management system. Corresponding to the agent oriented view of a business process, we model a workflow as a set of relating roles. The syntax of workflow model is given in EBNF as follows:

<Workflow_Model> ::= "WORKFLOW"
<Name>
[<Description>]
<Role_Models>
"END_WORKFLOW"
<Role_Models> ::= <Role_Model>{ "," <Role_Model>}

A role defines a prototypical function of an agent in a workflow. We adopt the concept of role basically from role theory [5], where a role is defined as a collection of duties and rights. *A role* refers to the expected behavior patterns an agent must perform. Duties of roles are modeled as obligations which specify what activities an agent must or must not perform on a set of target objects. Rights are modeled as permissions which specify what activities an agent is permitted (or forbidden) to perform on a set of target resources. Resource is something passively utilized during activity performance. These aspects are captured in the attributes of roles known as Obligation and Permission. A role model is defined in EBNF as:

<Role_Model> ::= "ROLE"
<Name>
[<Description>]
<Goals>
<Qualifications>
<Relationships>
<Obligations>

<Concurrency_Constraints>
<Permissions>
<Protocols>
"END_ROLE"

The attributes of a role has the following meaning:

Name: a name distinguishing the role from others.

Description: a brief English description of functions of the role in a workflow.

Goals: One or more goals the agent playing the role is responsible for.

Qualifications: One or more necessary pre-conditions or skills required to achieve the goals.

Relationships: one or more relations with other roles in a workflow.

Obligations: a set of obligations.

Concurrency constraints: an obligation expression specifying the mandatory sequencing, synchronization and permitted parallelism of obligations.

Permissions: a set of permissions or prohibitions.

Protocols: a set of protocols in which the role plays a part to achieve the goals.

Roles are defined separately from agent and from the organizational structure. One or more roles may be assigned to an agent. Agents are selected to play the roles according organizational policies and their capabilities. This link is done through the qualifications attribute of a role. And a qualification is defined as:

<Qualification> ::= "QUALIFICATION"
<Name>
[<Description>]
<COMMENT> | <Conditions>
"END_QUALIFICATION"

For an example, a reviewer role for an article on "Agent Oriented Workflow Systems" should have expertise in "Intelligent Agent". It can be represented as:

QUALIFICATION
Q1 Reviewer.Expertise = "Intelligent Agent"
END_QUALIFICATION

Roles and their relationship enable many organizational structures to be reflected in the multi-agent system. The relations from one role to another are categorized along two dimensions that is important for agent decision making. They are authority and cooperation relations, i.e.,

<Relationship> ::= "RELATIONSHIP"
<Name>
[<Description>]
<Relation_Type>
<Related_Role>
"END_RELATIONSHIP"
<Relation_Type> ::= "AUTHORIZE"
| "COOPERATE_WITH"
| "IS_AUTHORIZED_BY"

The authority relationship refers to the control relationship that exists between two roles, e.g., an editor has one AUTHORITY relation to an author. It is represented as:

RELATIONSHIP
R1 AUTHORIZE Author
END_RELATIONSHIP

2.2 Define Obligations of A Role

When fulfilling a role, an agent becomes subject to permissions, obligations and prohibitions by delegation.

An obligation is a prescription that a particular behavior is required. An obligation is fulfilled by the occurrence of the prescribed behavior [13]. It defines the functionality of a role. For examples, some obligations of an editor are : call for paper, select reviewers for a submission, collect reviews, make a review decision, etc. An obligation usually has an attribute "cost". If an obligation is violated then its player should be punished by a fine specified in its attribute "cost". The EBNF syntax of obligations is:

<Obligation> ::="OBLIGATION"
<Name>
[<Description>]
<Obligated_Roles>
<Authority_Roles>
<Benefited_Roles>
<Trigger>
<Actions>
<Constraints>
[<Cost>]
"END_OBLIGATION"

A permission is a prescription that a particular behavior is allowed to occur. A permission is equivalent to there being no obligation for the behavior not to occur. For example, the editor reserves the right to make a publication decision based on his/her own assessment of the best interests of the journal and its readers. *A prohibition* is a prescription that a particular behavior must not occur. A prohibition is equivalent to there being an obligation for the behavior not to occur. For examples, one prohibition of reviewers is: The article under review is a confidential document not yet accepted for publication. It should not be shown to or discussed with colleagues, cited as a reference or used in your own work.

In a role definition, the concurrency constraint specifies the mandatory sequencing, synchronization and parallelism of obligated activities. The basic operators are shown in Table 1, where O1 and O2 are obligations.

Table 1: Obligation Operators

O1 ; O2	; denotes sequential composition
O1 | O2	| denotes non-deterministic choice
O1 & O2	& denotes parallel composition

[O1]	[] denotes O1 is optional
<O1>	<> denotes O1 is necessary
O1*	* denotes O1 occurs 0 or more times
O1+	+ denotes O1 occurs 1 or more times

2.3 Model Roles Interaction Protocol

Ongoing interactions between roles often fall into typical patterns. These typical patterns of message exchange are called protocols. *A protocol* is specified by a set of rules governing the conversation among agents. A conversation is a set of interactions in order to ask for a service or request or update information. *An interaction* is a simple interchange of messages. It has the following attributes: speech-act, agent communication language, knowledge representation language, synchronization, sender, receiver, and ingredients[1].

Role interaction protocols can be graphically modeled by means of state-based techniques like Documentary Petri-Net [4], Role Activity Diagrams (RADs)[16], and Finite State Machine (FSM) [1]. A protocol is defined in EBNF as:

<Protocol> ::= "PROTOCOL"
<Name>
[<Description>]
<Initiators>
<Responders>
<States>
<Transition_Rules>
<Speech_Acts>
"END_PROTOCOL"

The attributes have the following meaning:

Name: a name distinguishing the protocol from others

Description: brief English description of its functions in a workflow;

Initiators: the role(s) responsible for starting the conversation;

Responders: the role(s) with which the initiator interacts;

States: a set of states of a FSM represent the states a conversation can be in. There is a distinguished initial state any conversation starts in, and a terminating state that when reached signals the termination of the conversation.

Transition rules: these rules control the workflow between these states.

Speech acts: a set of speech acts used in the protocol.

2.4 The Conceptual Architecture

To illustrate these definitions we present the conceptual architecture of agent oriented and roles based workflow in Figure 1. The architecture builds upon the general ideas introduced above. It lays out the workflow conceptual entities and their relationship.

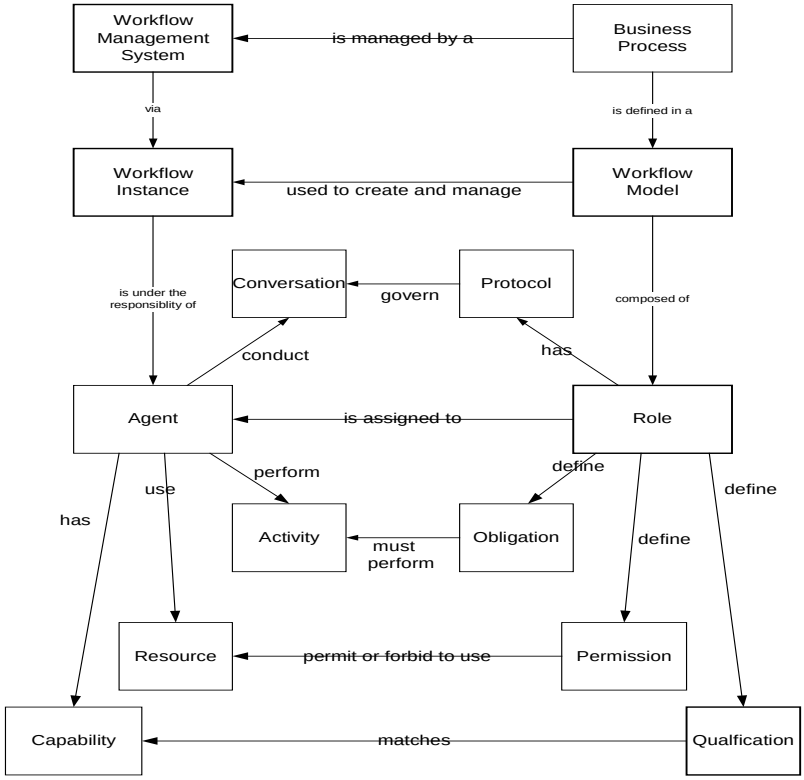

Figure 1 Relationship Between Basic Terminology of Agent Oriented and Role Based Workflow Systems

These concepts are subject to the following structuring rules:

- Workflow instance is a process combined with an organization.
- Agents comprising the organization
- Workflow is modeled as a set of relating roles.
- Roles and their relationships comprising the organizational structure;
- Roles played by agents;
- Obligated activities undertaken by agents;
- Permissions and prohibitions governing the creation, usage and deletion of resources by agents fulfilling roles;
- Protocol governing conversation (a set of interactions) between agents fulfilling roles;

The next section is dedicated to an approach for developing agent oriented and role based workflow management systems. It aims at bridging the gaps between the conceptual architecture and practical applications.

3 An Approach For Developing Agent Oriented And Role Based Workflow Systems

Despite the great interest around the agent technology in the scientific community, and the introduction of terms such as Agent-Based Software Engineering, there has been little work for defining a methodology for designing agent based workflow system. Based on the above conceptual framework, we try to outline an approach for developing agent oriented and role based workflow management system. Preliminarily the proposed approach follows three phases.

3.1 Phase 1- Role Based Analysis Phase

This phase includes three steps.

Step 1: Workflow goals identification and decomposition

In this step, the problem is what goals a workflow has and how they will be achieved. There are many methods used to refine goals and decompose them into sub-goals, e.g., scenarios analysis, bottom-up method, top-down method, hybrid of bottom-up and top-down method. Sub-goals can be seen as activities. There are two kinds of activities: elementary activity and complex activity. A complex activity has sub-activities and elementary activity can not be broken down any longer. A complex activity can be broken into its components, and components are broken down into sub-components until they become elementary activities.

Step 2: Roles determination and specification

Instead of assigning goals or activities directly to an agent, we specify that a goal or an activity is the responsibility or obligation of a particular role. A role defines a prototypical function of an agent in a workflow. The process of determining which goals or activities should be achieved by one role is role determination. Roles can be either:

Elementary roles which consist of collections of complex tasks or obligations

Roles which are a collection of elementary roles

Aggregate Roles, which are a collection of roles and coincide with social positions inside the organization. A position usually covers a number of roles. Agent occupy positions and play roles[25]. To keep our conceptual framework as concise as possible we do not employ position concept.

Once roles have been determined, their qualifications, relationships, obligations, permissions and constraints must be specified.

Step 3: Model and specify inter-role protocols and consistency checking

Protocols are modeled by using Formal Description Techniques like Finite States Machines so that the inconsistency of the rules can be checked.

When a protocol is realized in active database management system, a state transition diagram can directly be mapped into Event-Condition-Action (ECA) rules[2]. An ECA rule looks like:

ON <Event> IF <Conditions> THEN <Actions>

The semantics of a ECA rule is : when ECA-events occur, ECA-conditions are evaluated; and, if these conditions are evaluated to true, ECA-actions are executed. A set of ECA rules specify how an agent in a given state receives a messages of specified type, does local actions (e.g., updating local data), sends out messages, and switches to another state. The use of ECA rules rather than the standard production rules is important because ECA rules have a clear distinction between events and conditions.

Moreover, the interaction messages in the transition rules must be detailed in this step. The messages exchanged are represented as performatives (KQML speech acts)[8]. The content level of performatives is not part of the negotiation protocol, but determines the course of an individual negotiation as it is used in decision-making of agents.

The output of the phase is a specification of all roles and protocols involved in a workflow model.

3.2 Phase 2 - Agent Oriented Design Phase

This phase includes two steps.

Step 4: Identify agent-types and assign roles to agent-types.

With the introduction of roles, we now have to address the problem of how to specify “who can do what”. That is , what is the set of activities that an agent is allowed to perform as a workflow participant. We must identify agent-types and assign roles to agent-types.

In agent oriented workflow management system, we distinguish three general agent-types: Personal Agents, Actor Agents, Internal WfMS Agents. A personal agent acts on behalf of a participant, e.g., editor agent, reviewer agent, etc. An actor agent takes the role of an actor and is invoked by a workflow engine, e.g., a communication agent which takes the role of an email system in a WfMS. Internal agents of WFMS provide different views and access rights during workflow build-time and run-time. For examples, with respect to workflow execution we have identified these specific agent types for workflow manager, workflow initiator, workflow administrator, etc.

Step 5: Finish designing the agent-types

The roles and protocols provide a high-level information and we must provide additional detail for each agent-type to complete construction of the agents during implementation. Actually how an agent realize its functionality and obligations is beyond the scope of the approach.

The output of this phase is a detail specification of all agent-types in a WfMS. It lays the essential basis for implementing the Multi-agent system.

3.3 Phase 3 - Agent Oriented Implementation Phase (Step 6)

This phase focuses on implementation of the agent oriented workflow system on a suitable agent platform. Although many agent platforms are available such as

COOL[1] and AgentBuilder [22], currently none of them supports workflow management. Extra efforts have to be made and some issues are addressed in [28].

In the next section, we discuss only the first phase in detail by applying the approach in modeling the Internet Peer Review Process of Journal Articles.

4. Case Study: Modeling The Internet Peer Review Process

4.1 The Internet Peer Review Process of Journal Articles

The Internet offers any individual the ability to publish and exchange information. The project *NetAcademy* (http://www.netacademy.org/) aims to provide an Internet based platform for scientific community to aid the creation, exchanging and dissemination of scientific knowledge[11]. The International Journal "Electronic Markets" is a part of the NetAcademy. A World Wide Web site has been created for the Journal (http://www.electronicmarkets.org/). Here, the online Journal publishes research articles on Electronic Markets that have been accepted for print publication. The full text of all documents will be indexed and an online search facility Q-Search Engine is provided.

Essentially, the peer review process is a quality control mechanism for the academic journal. Editorial peer review in the past has been secretive (because only the Editor can see the whole process), narrow (because each article is usually reviewed by only two or three people) and somewhat arbitrary (because the choice of reviewers is limited by the Editor's knowledge and the time constraints of the potential reviewers) [3].

Consider the paper peer review process. We assume that the review process is to select reviewers, distribute the paper(s) to the selected reviewers, have the reviewers perform the reviews and collaborate in producing a joint review document, and finally forward it to the authors. This is an ad hoc workflow because it involves: (1) negotiation for selecting the reviewers, and (2) collaboration between the reviewers for producing a joints review.

This *ad hoc workflow* tasks involve human coordination, collaboration, or co-decision. Thus, the ordering and coordination of tasks are not automated but are instead controlled by humans. Furthermore, the task ordering and coordination decisions are made while the workflow is performed. Ad hoc workflows involve small teams of professionals and are intended to support short term activities which require a rapid workflow solution. Workflow Management Systems (WFMS) that support ad hoc workflow must provide functionality for facilitating human coordination, collaboration, and co-decision. The infrastructure technology currently used by ad hoc WFMSs ranges from "enhanced" electronic mail to group calendaring and conferencing systems [10].

As discussed in section 1, the use of agent technology for this domain will be motivated. We intend to develop an online Internet peer review and publish system as

a multi-agent system based on World Wide Web. The submission is placed on the Journal web site. Executive editor invites authors, selected reviewers and copyeditor to participate in the Online discussion [3]. Access by password is restricted to these participants. Discussion is conducted on the first submission, the authors' revision, copyedited article. Editor concludes with the editor's decision to reject, or definitive instructions for revising the article. Reviewers post their reviews to the discussion list associated with the article. Authors observe the discussion of their paper and can respond to comments with explanations.

We are about to handle the reviewer selection, communication and the commitments automatically through agent technology. To do this, the author, editor, and each referee run a software agent which enable them to participate in this distributed peer review procedure. The procedure integrates peer review with the electronic publication and workflow management, allowing rapid publication after rigorous peer review. It is obvious the cooperation and communication among editor, authors and reviewers are enhanced and improved. It preserves the chief virtue of the traditional peer review system (expert appraisal before publication), but brings a new openness to the procedure that we hope will enhance its accuracy and fairness. Therefore, the review quality and major delay time will be guaranteed and improved.

4.2 Roles Determination And Specification

The goal of the Peer Review Process is to control the quality of a publication. It may be decomposed into these sub-goals: submit an article, receive an article, find three reviewers for an article, distribute the article to the three reviewers, three reviewers review the article, collect the reviews, make a review decision, forward the decision to the author, revise the article, copyediting the article, etc.

If we assign one sub-goal to one elementary role, then the same numbers of elementary roles as that of sub-goals can be determined. Now what we need to do is specify these elementary roles. As an example, we model the role "reviewer-selector" in Table 2.

Table 2: Specification of Role "Reviewer-selector"

Attribute	Value
Name	Reviewer-selector
Goals	G1: appoint three reviewers for a submission
Qualifications	C1: Know a set of potential reviewers C2: Have a submission for a peer review C3: Be able to match the content of an article with expertise of reviewers
Relationships	R1 IS_AUTHORIZED_BY editor R2 COOPERATE_WITH potential-reviewer R3 AUTHORIZE committed-reviewer
Obligations	O1: initial call for review O2: allocate a review-time O3: find a potential reviewer

	O4: negotiate with a potential reviewer O5: appoint a potential reviewer as a reviewer
Concurrency Constraints	O1 ; [O2 & O3 ; O4]* ; O5+
Permissions	PM1: Access submissions and reviewers database PM2: Free to contact with the potential reviewers PM3: Allocate a review-time PM4: Cancel a review request PM5: One should not review his own submission
Protocols	PT1: Initial-Call-for-Review PT2: Negotiation with a potential reviewer

The role "Reviewer-selector" has 5 obligations. The obligation "review-time allocation" is modeled in Table 3.

Table 3 Obligation "review-time allocation" of role "Reviewer-selector"

Attribute	Value
Name	Reviewer-selection
Obligated roles	Reviewer-selector
Authority roles	Editor
Benefited roles	Author
Trigger	Insufficient reviewers accept the initial terms to review an article
Actions	Choose a potential reviewer for the article Allocate a review-time to the potential reviewer
Constraints	<= 4 weeks
Cost	middle

4.3 Model And Specify Role Interaction Protocols

Let us consider the sub-process of selecting the reviewers in detail [21]. Suppose the editor aims to ensure quality by having each submission reviewed by three referees. Authors submit their papers to editor, complete with paper identifier (authors, title, etc.), an abstract and a set of keywords. The editor then broadcasts to the network of referees that it has a review job to do with a proposed resource allocation of 3 weeks, and invites bids to undertake the work. If enough reviewers respond favorably, then three of them are chosen and are appointed.

The protocol " Initial Call For Review" between reviewer-selector and a set of N potential reviewers is modeled by FSM in Figure 2.

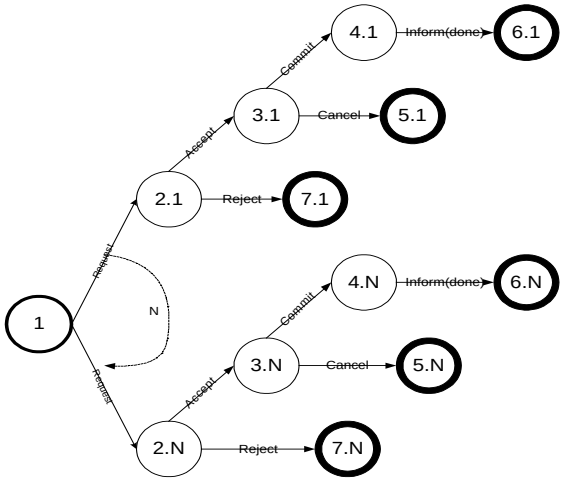

Figure 2 Initial-Call-For-Review Protocol

If insufficient reviewers accept the initial terms, then the editor selects what it considers to be a set of appropriate referees based on the referee's areas of expertise and the submission's keywords, and engages in one-to-one negotiation with each in the attempt to get a commitment to review the submission. At first the editor sends a proposal to potential reviewers. The terms of the proposal maybe "weak" than the last request. The receiver may reject it, accept it, or provide a new offer to the editor. This continues until either enough reviewers have been found or the set of potential reviewers has been exhausted. In the latter case an alert is raised.

This negotiation protocol between the reviewer-selector and a potential reviewer is modeled in Figure 3.

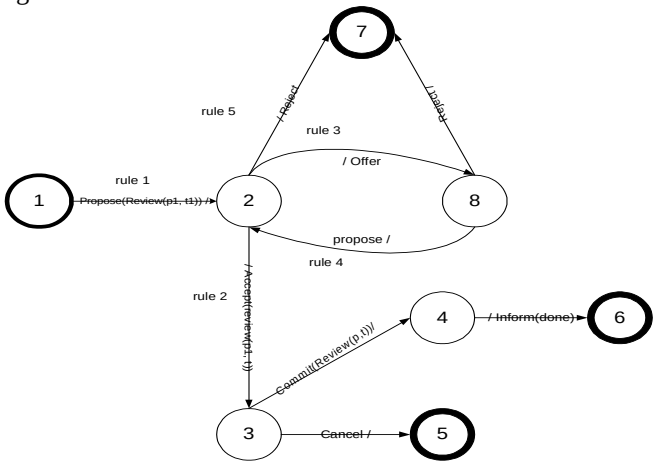

Figure 3 Negotiation Protocol

For examples, the negotiation protocol is specified as follows:

Name: Negotiation protocol.

Description: Reviewer-selector negotiates with a potential reviewer to set an agreement for reviewing an article

Initiator: reviewer-selector

Responder: a potential reviewer

There are 8 states in this state transition diagram:

State 1: Start state.

State 2: A potential reviewer received the proposal from the reviewer-selector.

State 3: The proposal has been accepted by the potential reviewer.

State 4: The reviewer-selector has appointed the potential reviewer as a reviewer and committed him to finish the review before the agreed deadline.

State 5: The reviewer-selector has cancelled the review proposal. END State.

State 6: The review has been finished and the reviewer-selector has been informed about it by the reviewer. END State.

State 7: The potential reviewer has rejected the proposal. Or the reviewer-selector has rejected the offer from the potential reviewer. END State.

State 8: The reviewer-selector has received an offer from the potential reviewer.

Now let's see the transition rules. If we map one transition into one ECA rule, then we have the following ECA rules of the negotiation protocol:

Rule1 (State 1 ---> State 2):

ON there are insufficient reviewers accept the initial review terms

IF reviewer-selector found a possible reviewer and he has made a proposal for negotiation

THEN he sends the review proposal to the possible reviewer

Rule2 (State 2 ---> State 3):

ON the potential reviewer receives the proposal for review,

IF he decides to accept the proposal

THEN he sends an accept-message to the selector in reply with the proposal.

Rule3 (State 2 ---> State 8):

ON the potential reviewer receives the review proposal

IF he decides to provide an offer to the selector

THEN he sends an offer-message to the selector in reply with the proposal.

Rule4 (State 8 ---> State 2):

ON the selector receives an offer from the potential reviewer

IF he decides to make a new proposal

THEN he sends a new proposal to the potential reviewer in reply with the offer.

Rule5 (State 2 ---> State 7):

ON the potential reviewer receives the review proposal

IF he decides not to accept the request

THEN he sends a reject-message to the selector in reply with the proposal.

Rule 6 ...

The above negotiation protocol encompasses primitives from the most common phase of negotiation such as Propose, Offer, Accept, Commit, Reject, Cancel, Inform, etc. In the Initial-Call-For-Review Protocol, the primitive "Request" is used. A

“Request” to do A is non-negotiable: a negotiable request is a “Propose”, which allows the receiving agent to make a counter-offer using the “Offer” message. A high level deontic specification of the protocol can be found in [21].

Following is an example of a “Propose” sent from the reviewer-selector to a potential reviewer asking for reviewing an article titled ‘title1’ within 3 weeks.

(propose
:sender ?reviewer-selector
:receiver ?potential-reviewer
:language predicate-logic
:ontology NetAcademy
:content review(?potential-reviewer, article1) ^ has(article1, title1, abstract1, keywords1) ^ has-review-time(article1, 3-weeks))

The ontology is a formal description of the problem domain. It maps these symbols to a well-understood meaning for the problem domain. For one agent to properly understand the meaning of a message from another agent, both agents must ascribe the same meaning to the symbols (constants) used in that message.

4.4 Identify Agent-Types And Assign Roles to Agent-Types

As discussed in section 3.2, we distinguish among Personal agents, Actor agents and Internal WfMS agents. For examples, we assign the role “deliver messages” to an actor agent that may be invoked by the workflow engine and invoke an existing email system. The internal workflow agents are not addressed here but in [28]. Following we discuss the main functions of some personal agent types in the Internet peer review process.

By assigning the elementary roles into agent types, we have identified these personal agent types: author-agent type, editor-agent type, reviewer-agent type, etc. This arises very naturally from the organizational roles involved in a peer review process.

Author agent: It manipulates author’s work items. It initializes a peer review process when it submits an article from the author to the editor-agent. It alerts the author when it receives a rejection or revision request message from editor-agent, or when any new comments have been made on his article. It deliveries author’s revision version and comments to editor-agent in the online review discussion, etc.

Editor agent: It manipulates editor’s work items. Responsible for receiving the coming messages to the editor. Alert the editor to make an initial decision when a submission has been received. If the editor tells him “Yes”, then it selects three reviewers for the submission. If there are insufficient accepted reviewers, it continues to negotiate with potential reviewers’ agents to reach an agreement for a review. Other functions are: Collect the committed reviews and publish it on the web; Maintain the potential reviewers database; Alert the editor to make a review decision when it receives all reviews; Remind the editor of the deadline for an open review, etc.

Reviewer Agent: Monitor and alert reviewer if a coming message arrives; Reply a review request; Negotiate with editor agent to reach an agreement for a review;

Remind reviewer of a deadline for a committed review; Transfer review report to the editor agent; etc.

In this section we have studied the Internet Peer Review Process by applying the role based modeling approach proposed in a previous section. We continue to address some issues on the agent oriented design and implementation in [28].

5. Related Work

In the *BPR and workflow management community*, researchers are developing new kinds of workflow which meets the requirements of dynamic and volatile business environment. Eillis et al choose people and goals as a starting point for modeling office workflow instead of choosing procedures and activities[7]. They suggested an extended ICN (Information Control Nets) model which includes the concept of goals. Bussler and Jablonski have proposed to integrate organisational model, activity model or data model in the workflow management environment [6]. Yu and Mylopoulos proposed a framework which focuses on modeling of strategic actor relationships for a richer conceptual model of business processes in their organisational setting [25]. They view organizations as being made up of social actors who are intentional – have motivations, wants, and beliefs and strategic – they evaluate their relationships to each other in terms of opportunities and vulnerabilities. These correspond to our goal-achieving perspective and agent view of a business process.

Action Workflow follows that a business process should be viewed as a network of commitments [20]. It starts from the language/action perspective of the *speech acts theory* which severs also as a theoretic foundation of Agent Communication Languages [8]. RADs are a powerful modeling technique which use a number of notational primitives to express complex process behavior. In *Role Activity Diagrams (RADs)*, a role has been described as being "... comprised of a set of activities which taken together, achieve some particular goals" [16]. In addition, role concept has been studied in distributed systems management [18], Open-EDI [4] [14] , the enterprise viewpoint of Open Distributed Processing [13], and Media Theory [17].

Whereas organization has been presented as a major issue of multi-agent systems, very few works have attempted to develop models of such systems using organizational approaches. There is no role concept in ADEPT[15]. Concepts such as roles, teams or organizations are used in a very informal, almost casual, way [12].

Our methodology distinguishes itself from others in that it takes an agent oriented view of a workflow. It is a very natural and an easy way to model workflow by our role model which matches Agent Oriented Programming [22][23] very well. It can integrate seamlessly the role based analysis with agent oriented design and lead to an effective implementation of agent based workflow systems. It is an effort to bridge the gaps between Workflow Community and Agent Community.

6. Conclusion and Future Work

This paper explores an innovative conceptual framework for agent oriented and role based workflow management that is novel and exhibits several advantage over previously approaches. A workflow is viewed as a collection of autonomous, problem solving agents which interact with others when they have interdependencies. We model a workflow as a set of relating roles. A role has these essential features: goals, obligations, permissions, protocols, qualifications, etc. Protocols govern the interaction among roles and conversation among agents. We take the view that coordination of workflow is achieved by agent communication. This goes one step further than current generation workflow systems which achieve coordination of activities by highlighting their interdependencies. Agent oriented workflow takes a distributed, and hence more robust, flexible, and scaleable approach for workflow management. The vision of role based analysis and agent oriented design establishes the framework for the practical work. Many similar Internet–enabled workflow applications in Electronic Commerce are easily imaged.

In particular, because we take role as an important mechanism for building organizations of agents, the workflow modeling procedure is similar to that of organization design. Using roles for modeling workflow is valuable in four ways: independence between individuals and definition of the process; to provide a way to balance distribution of work; to provide an abstraction of agent; to integrate role based analysis and agent oriented implementation perfectly.

However, managing workflow in a Multi-agent framework is a fertile area for research and development. There are significant problems to be solved to ensure its future success. First of all, we are about to implement the Internet peer review system in a Multi-agent framework as a part of the project NetAcademy [28]. We are formalizing the conceptual framework and developing a Role Based Workflow Description Language. Moreover, we are about to employ deontic logic to represent and reason about obligations of agents to integrate our role model and the extended BDI agent architecture seamlessly[1]. Finally, the approach for developing agent oriented and role based workflow management systems must be evaluated and improved through applications.

Acknowledgements. We thank Dr. Ulrike Lechner and anonymous referees for their useful comments and suggestions.

References

1. M. Barbuceanu. “How To Make Your Agents Fulfil Their Obligations”. In Proceedings of The Third International Conference on the Practical Application of Agent and Multiagent Technology, PAAM’98, 1998
2. M. Berndtsson. “Active Capability Support For Cooperative Strategies in Cooperative Information Systems”, Ph.D. Dissertation, Faculty of Science, University of Exeter, UK, July 1998

3. C. Bingham, C. Higgins, R. Coleman, M. B Van Der Weyden. “The Medical Journal of Australia Internet peer-review study”. The LANCET, Vol. 352, August 8, 1998, pp441-445
4. R. W.H. Bons, R. M. Lee, R. W. Wagenaar and C. D. Wrigley. “Modeling Inter-organizational Trade Procedures Using Documentary Petri Nets”, Proceedings of the Hawaii International Conference on System Sciences 95, Hawaii,1995
5. B. J. Biddle and E. J. Thomas (Editors): Role Theory : Concepts and Research, John Wiley & Son, Inc. 1966
6. C. Bussler and S. Jablonski. “An Approach to Integrate Workflow Modeling and Organization Modeling in an Enterprise”, Proceeding of third Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, April 17-19, 1994, Morgantown, West Virginia
7. C. A. Ellis and J. Wainer. “Goal-based models of collaboration.” Collaborative Computing, 1(1), 61-86, (1994).
8. T. Finn, Y. Labrou and J. Mayfield. KQML as an agent communication language. Chapter in “Software Agents”, edited by Jeff Bradshaw, MIT Press, Cambridge, 1997
9. M. Fox and M. Barbuceanu. “An Organisational Ontology for Enterprise Modeling: Primary Concepts for Linking Structure and Behavior”. Computer in Industry, Vol. 29, pp123-134 , 1996
10. D. Georgakopoulos, M. Hornick and A. Sheth. “An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure”. Distributed and Parallel Database. 3, 119-153, 1995, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
11. S. Handschuh, U. Lechner, D. Lincke, B. F. Schmid, P. Schubert, D. Selz, and K. Stanoevska-Slabeva. “The NetAcademy - A New Concept for Online Publishing and Knowledge Management”. in Proceedings of the International. Workshop on Advanced Communication Services (ACoS'98), Lecture Notes in Computer Science 1385, pp. 29-43. Springer-Verlag, 1998.
12. I. Hawryszkiewycz and J. Debenham. “A Workflow System Based on Agents”. Proceedings of 9th International Conference on Database and Expert Systems Applications; DEXA’98, Vienna, Austria, Lecture notes in computer science; Vol. 1460, Springer, August 1998
13. ISO/IE DIS 10746. Open Distributed Processing - Reference Model (RM-ODP), ISO/IE DIS 10746-2 and 3, 1996
14. ISO/IEC 14662. Information Technologies – Open-edi reference model, International Standard, Reference number: ISO/IEC 14662:1997(E)
15. N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, M. E. Wiegand: “Agent-based Business Process Management”. International Journal of Cooperative Information Systems, 5 (2&3), 1996, 105-130.
16. P. Kawalek. “Comments On The Use Of RADs In Case Studies”. IOPener Vol.2, No.4, July 1994, Praxis Systems, Bath, UK.
17. U. Lechner and B. F. Schmid. “Logic for Media – The Computational Media Metaphor”. Proceedings of the 32nd Annual Hawaii International Conference on System Science, HICSS’99, Hawaii, Jan 5-8, 1999.
18. E. C. Lupu. “Role-Based Framework for Distributed Systems Management”. Ph.D. Thesis, Department of Computing, Imperial College of Science, Technology and Medicine, University of London, July 1998
19. T. W. Malone and K. Crowston. “The Interdisciplinary Study of Coordination”. ACM Computing Surveys, Vol 26, No. 1, March, 1994, p87-119

20. R. Medina-Mora, T. Winograd, R. Flores, F. Flores. “The Action Workflow Approach to Workflow Management Technology”. Proceedings of CSCW 92. November 1992. pp281-288
21. J. Pitt, M. Anderton and J. Cunningham. “Normalized Interactions Between Autonomous Agents: A Case Study In Inter-organizational Project Management”, Proceedings of the International Workshop on the Design of Cooperative Systems. January 1995, Antibes-Juan-les-Pins, France, pp76-95
22. Reticular Systems, Inc. “AgentBuilder – An Integrated Toolkit for Constructing Intelligent Software Agents”. User’s Guide, Version 1.1, Reticular Systems, Inc. , 4715 Viewridge Avenue, Suit #200, San Diego, California 92123, January 19, 1999. http://www.agentbuilder.com
23. Y. Shoham. “Agent-oriented Programming”, Artificial Intelligence 60 (1993) pp51-92
24. WfMC. Workflow Management Coalition Terminology & Glossary, WFMC-TC-1011, Document Status- Issue 2.0, June 1996
25. E. S. K. Yu and J. Mylopoulos. “Modeling Organizational Issues for Enterprise Integration”. Proceedings of International Conference on Enterprise Integration and Modeling Technology, Turin, Italy, October 28-30 1997
26. L. Yu: “A Coordination-based Approach for Modeling Office Workflow”, in Business Process Modeling, edited by Bernd Scholz-Reiter and Eberhard Stickel, Springer Verlag, 1996, pp. 250-65
27. L. Yu, K. Stanoevska-Slabeva and B. F. Schmid. “Supporting Distributed Corporate Planning through New Coordination Technologies”. Proceedings of the International Workshop on Coordination Technologies, in conjunction with 9th International Conference on Database and Expert Systems Applications, DEXA'98, Vienna, August 24-28, 1998
28. L. Yu and B. F. Schmid. “The NetAcademy – Managing Internet Peer Review Process in a Multi-agent Framework”. Presented at IJCAI Workshop on Intelligent Workflow and Process Management: the New Frontier for AI in Business. August 1-2, 1999, Stockholm, Sweden.