Electronic Commerce Based on Mobile Agents in Distributed Application

Whe-Dar Lin
Department of Information Management
Overseas Chinese College of Commerce

Abstract

Electronic commerce based on mobile agents in distributed application is expanding from the simple notion of electronic store to the notion of virtual enterprises. A distributed transport protocol to provide concurrent support for different real time and electronic commerce applications on transport protocol is proposed. The electronic commerce based on mobile agents in distributed application transport protocol is specified and its performance is derived within the field of quality of service support for distributed electronic commerce systems by mobile agents.

This paper presents electronic commerce based on mobile agents in distributed application transport protocol architecture for distributed electronic commerce systems which includes a general protocol layer for the transport of electronic commerce objects. The proposed protocols are based on specify the synchronisation requirements of electronic commerce applications. After presenting transport protocol shown how we have been used to design the general electronic commerce architecture whose transport layer is based on the concept of a partial order connection, an extension of the classical connection oriented as TCP and connectionless as UDP transmission concepts. The most of the work has been within tile context of individual architectural layers such as the distributed system platform, operating system and network.

Mobile agent effect of the real time protocol on data users is evaluated and it is shown that while real time users obtain delay constrain service, the total channel utilization is also improved when compared to a standard network in which both data and real time users access the channel. We proposed the development of electronic commerce based on mobile agents in distributed application architectures which incorporate quality of service configurable interfaces and quality of service driven control and management mechanisms transport architectural layer.

Keywords: Mobile agent, Distributed electronic commerce systems; Electronic commerce information; Electronic commerce protocols, Virtual enterprise;

1. Introduction

Electronic commerce based on mobile agents in distributed application need for supporting transmission of real time or periodic type traffic on communication networks has gained large attention. Local area network systems, in general, have been traditionally viewed as being incapable of carrying electronic commerce and high speed real time traffic due to data rate and protocol limitations. However, the emerging high speed Ethernet versions, on the other data rate demands and resources seem to match quite well for many practical scenarios. Therefore, appropriate integrated transport protocols are needed that (i) support several classes of real time traffic and (ii) compatible with the original data only protocol so that on an integrated electronic commerce services stations with integrated functionality can coexists with standard data only stations.[1][2][4][5]

Mobile agents, in the past, a few such integrated protocols have been proposed, analysis and implemented to support a single-class application. [3][8][10][11][16] Mobile agents Meeting quality of service guarantees in distributed electronic commerce systems is fundamentally an end-to-end issue, that is, from application-to-application. Consider, for example, the remote user in the distributed system platform, quality of service assurances should apply to the complete row from the remote server across the network to the points of delivery. This generally requires end-to-end admission testing and resource reservation in the instance, followed by careful co-ordination of disk and thread scheduling in the end-system, packet/cell scheduling and flow control in the network, and finally active monitoring and maintenance of the delivered quality of service. A key observation is that for applications on the transfer of electronic commerce, and in particular continuous flows, it is essential that quality of service is configurable, predictable and maintainable system, including the end-system devices, communications subsystem and networks. Furthermore, it is also important that all end-to-end elements of distributed systems architecture work to achieve the desired application level behavior.[7][11][13][15]

The last phase led to the general electronic commerce architecture that has been design to the network and end-system communications resources, the end-to-end delay, and to fulfill the electronic commerce temporal requirements. [9][10][12][14][16]

These ideas were implemented in the contest of a generic system and the related general architecture will finally be discussed.

2 The Electronic Commerce Mobile Workflow Agent

In the environment described so far, useful information and services necessary to achieve a business goal reside at various servers of different enterprises that need to be dynamically linked to form a VE. In our framework, the task of linking the sites and hence their relevant services forming a multi-organizational workflow is accomplished by mobile agents called workflow agents. Agents be dynamically dispatched and can roam from one server to another based on their mission and acquire new knowledge from the visited servers. The information carried in the agents may in turn spawn other agents or terminate previously created agents. In this section, we describe how agents identify the most appropriate sites, glean the relevant information front the huge data repositories at these sites and employ the services available at these sites to complete the objective of the dynamic workflow in a resource efficient and cost effective manner.

One has to describe the different QoSs of complex electronic commerce objects and in particular their synchronization requirements in the temporal requirements between the different media sub-objects. It will also be shown that it is simple and easy to relate these logical and time requirements to electronic commerce presentations.

2.1 Agent System Architecture

Each site in the environment can be both a service requester and a service provider and maintains a service or information agent. A service repository contains descriptions of agents offered services by its site and agents of other service providers and their services. We also assume the existence of trading serves which in addition to offered services, they store requested services in their repositories. Both provided and requested services we specified as workflow views. In accordance to their authentication, agents initiated by a site or by another site may use, modify and delete service descriptions

To achieve its function an agent is associated with a set of data that describe its mission and roaming and advertising strategies, its scope and authentication credentials. Depending on its type, the metadata elements may be specified by the user or derived from the application and possibly constructed as a set of conceptual relations. The user must restrict the roaming and advertising strategy to a specific domain. Similarly, the nature of the application may impose strict authentication and filtering policies to achieve stronger security and guard against potential intrusions. Formally, an agent can be defined as follows:

agent = (mission, profile, target, non-target, strategy)

The mission (< type) >) specifies the type of agent (service provider or service requester) and service advertisements. An ad is specified as a workflow view representing a service along with the associated set of preconditions and constraints. An agent can dynamically change its mission by decomposing a workflow view into several smaller views and delegating some of them to another agent or dropping them, or by expanding a workflow view incorporating opportunities, omitted activities and data. Whether or not an agent is able to change its mission depends on the profile of the agent. Duration with each transition or with each place of the net, in the latter case, which is the one considered here, this value indicates the time a token has to remain in the place. The electronic commerce object produced on site A must be sent to site B, through a wide area network, using such protocols. Two main approaches have been proposed to design new electronic commerce communication systems.

The first one, called ILP, is based on the use of a simple network protocol, such as IP. On top of IP, the electronic commerce protocols needed by a given application are implemented in the application code, e.g. at the user level. This allows advanced application developers to define the most appropriate protocols for their specific applications and to optimally tune the corresponding software. Nevertheless, each user that implements and application has to develop or select the appropriate communication software: no common end-to-end service is provided to the users. We call this approach "Network-Aware Applications" (NAA) as each application directly uses in the best possible way, the existing network.

The second approach, takes a different view. It is based on the design of an advanced communication layer that integrates and handles, as much as possible, different electronic commerce requirements. This approach leads to a communication layer which is more general and more complex than the common UDP-TCP transport layer. The interest of this approach is that the application user software becomes much simpler to implement because the communication layer provides the advanced functions needed by the electronic commerce applications; the users do not need to directly design and implement them. We call this approach "Application-Aware Networking" (AAN). As will be seen, this second approach needs to generalize the common important present protocols, UDP and TCP, which become specific cases within a large set of protocols resulting from the AAN approach.

The profile specifies the behavior of an agent, first it describes its authentication credentials and identify its site and application of origination. Second it describes its itinerary and rules for identifying servers and for modifying its itinerary. These rules also include termination criteria. Third, it describes how the workflow views in the mission are manipulated. In the case of a service provider, these rules include dissemination criteria whereas in the case of service requester, these rules include acquisition criteria. Fourth, it describes cost comparison strategies to aid negotiation and rates that define negotiation requirements and contract establishment.

Agents can be categorized into two broad types based on the strategy parameter. These are pro-active versus reactive agents. Pro-active agents conduct groundwork in anticipation of the requirements of workflow that will be invoked in the future. For instance these agents could conduct a systematic search for secondary and tertiary sources or service providers. These information sources/service providers could be of different types. Note that pro-active agents have the potential of providing fast workflow execution if appropriate groundwork has been completed before the invocation of the workflow. Further, the pro-active agents may be programmed to be automatically re-launch their actions periodically to obtain up-to-date information about service availability, and log changes in the environment. The periodicity of these actions depend on the availability of computing and networking resources. Reactive agents on the other hand, are invoked only when a specific workflow is initiated. At any given time, the system could have a mix of reactive and pro-active agents depending on the availability of resources.

Agents may be created dynamically by a workflow and invoked, possibly repeatedly, during the execution of the workflow. In the following section, we describe the overall system architecture to support the agent-based framework-

3.1. Agent System Architecture

The system architecture to support agent-based dynamic workflow should address the basic issues of how to organize the resource space flexibly and dynamically to make the implementation of a large-scaled infrastructure feasible. The architecture must allow the system's structure and interactions between agents and servers to evolve over time in accordance with usage patterns.

Service advertising and agent roaming strategies, capture the interactions between agents and servers and promote service advertising, filtering, and information archiving in a user and context sensitive subgroups so as to reduce the amount of overhead required to locate a server and limit the interaction to particular groups of servers which are most likely to assist effectively in the execution of the workflow. Capabilities must be provided to allow the agents to plan their roaming itinerary in the most effective way by continuously seeking to construct virtual links among themselves and potential seven based on the metadata that describe the content and type of their mission. The advertising and recruiting strategies specified by the profile, and the context within which the workflow task is being undertaken. support negotiation between agents and servers so that their collaboration is achieved in the most effective way. To achieve these objectives, the proposed architecture naturally draws upon existing components of an internet work, including name servers, authentication servers and routers, and adds now components, namely agent managers, agent credential authenticators, and agent communication servers.

Agent managers at different networks cooperate to provide the basic mechanisms and capabilities to support the interaction between servers and agents. Agent authenticators verify and validate the credentials of the agents, such as the identity of the agent originator, the network address where the originator resides and possibly the name and address of author authorities sanctioning entities. The agent communication server handles communication and group formation among agents and servers to facilitate their cooperation toward efficiently executing the steps of the workflow. The overall structure and functional level decomposition of the proposed architecture, we briefly discuss the function and services provided by each of these components.

3.1.1. From UDP and TCP connections to POCs.

Let us now consider transport protocols to transfer electronic commerce objects. Electronic commerce objects consist of different media having different QoSs. For instance, some of the business may require fully reliable transmission, as for instance small parts of images or entire frames may be lost. Furthermore, from Section2, the requirements of the objects should be deduced from the previously given Petri net based models.

In a set of elements used in building QoS into distributed electronic commerce systems is described. This includes QoS principles which govern the construction of a generalized QoS framework, QoS specification which captures application-level QoS requirements, and QoS mechanisms which realize the desired application end-to-end QoS behavior.

The fundamental idea t this point is to notice that a Petri net is a partial order (PO), i.e. a PO can be derived from any Pertri net however, not all POs correspond to Petri nets. Then the second idea is to analyze UDP and TCP with respect to this PO. If the modes are adequate, i.e. if electronic commerce objects are partial orders, then different objects imply different partial orders. It follows that the set of objects can be related to the set of POs, which is a lattice having as extra no order and total order.

It is now easy to extend the previous formal models to give to users a way of defining their requirements in terms of reliability. It is for instance easy to characterize a place by a simple reliability figure, a Boolean, 0 or 1, meaning that the information related to this place can or cannot be lost. Of course more complex possibilities may be defined.

3.1.2. Agent System Architecture

The system architecture to support agent-based dynamic workflow should address the basic issues of how to organize the resource space flexibly and dynamically to make the implementation of a large-scaled infrastructure feasible. The architecture must:

· allow the system's structure and interactions between agents and servers to evolve over time in accordance with usage patterns, service advertising and agent roaming strategies,

· capture the interactions between agents and servers and promote service advertising, filtering, and information archiving in a user and context sensitive subgroups so as to reduce the amount of overhead required to locate a server and limit the interaction to particular groups of servers which are most likely to assist effectively in the execution of the workflow. Capabilities must be provided to allow the agents to plan their roaming itinerary in the most effective way by continuously seeking to construct virtual links among themselves and potential seven based on the metadata that describe the content and type of their mission. The advertising and recruiting strategies specified by the profile, and the context within which the workflow task is being undertaken. support negotiation between agents and servers so that their collaboration is achieved in the most effective way.

To achieve these objectives, the proposed architecture naturally draws upon existing components of an internet work, including name servers, authentication servers and routers, and adds now components, namely agent managers, agent credential authenticators, and agent communication servers.

Agent managers at different networks cooperate to provide the basic mechanisms and capabilities to support the interaction between servers and agents. Agent authenticators verify and validate the credentials of the agents, such as the identity of the agent originator, the network address where the originator resides and possibly the name and address of author authorities sanctioning entities. The agent communication server handles communication and group formation among agents and servers to facilitate their cooperation toward efficiently executing the steps of the workflow. The overall structure and functional level decomposition of the proposed architecture is depicted. In the following sections, we briefly discuss the function and services provided by each of these components.

3.1.3. The basic parameters

As a consequence, electronic commerce communications can take benefit of partial order but also of partial reliability provisions to improve their performance.

It appears that the untimed Petri net of these models (i.e. the Petri net obtained when deleting the time values) represents a partial order between the places, i.e. between the sub-objects of the electronic commerce object.

As there is a lattice of partial orders, a lattice of corresponding partial order protocols exist in the R-POC segment. It follows that the partial order of a protocol can be selected to be the one defined by the Petri net. The partial order defining the electronic commerce object precisely defines the partial order electronic commerce protocol for this object. As a first and fundamental consequence, electronic commerce POCs can be directly and easily linked to electronic commerce objects.

3.1.4. Basic behavior

A number of QoS principles motivate the design of a generalized QoS framework: integration principle states that QoS must be configurable, predictable and maintainable traverse resource modules over all architectural layers to meet end-to-end QoS. Flows as CPU, memory, electronic commerce devices, network, etc. at each layer from source media devices, down through the source protocol stack, across the network, up through the receiver protocol stack to the devices. Each resource module traversed must provide QoS configurability based on a QoS specification, resource guarantees provided by QoS control mechanisms and maintenance of ongoing flows;

It will now be shown that POs are able to properly transfer electronic commerce objects. To illustrate the interest of a transport POC, let us consider first PO together with full reliability. Let us assume that each place (to which a sub-object is associated) corresponds to a service data unit (SDU) to be sent by the user of the communication layer. For example if the sender will send the SDUs sequentially, using the sequence: 1; 2; 3; etc. Let us now assume that SDU 1 is lost. If a TCP CO (totally ordered reliable) protocol is used, then retransmission occurs and, using a go-back-N procedure, the following SDUs are not saved: the sequences have to be retransmitted, which uses bandwidth and increases delay. Note that if a go-back N procedure is not used, the SDUs have to be saved in the receiving transport entity a waiting ordered delivery to the user, which increases memory resources and also delays.

As shown sub-objects 1, 2, … n are not totally ordered. For instance, as 1 and 2 are in parallel, there is no required order between them: receiving 1 before 2 or 2 before 1 is acceptable, without any consequence to the user.

If the receiving entity knows that places 1 to n are in parallel. It is able to know that these SDUs can be delivered in any order: if SDU 1 is lost, SDUs 2 to n can be delivered as soon as they are received instead of being saved ore retransmitted again.

This is precisely what partial order protocols do and it follows that POCs need less bandwith reduce the amount to buffers and furthermore give shorter delays to the applications. Note that this approach is very general, as it is based on a formal modeling of the electronic commerce objects.

We note that Pos can be implemented in the NAA approach or using the AAN approach. The layered AAN approach will be considered there, where the POCs provide to applications a service is application independent, but can be easily parameterised by passing the required partial order as an application parameter.

Of course, in any case, such and approach needs to define the requirements in terms of a well defined, application dependent formal model, to make the requirements very precise.

3.1.5. Method

The separation principle states that media transfer, control and management are functionally distinct transport protocols architectural activities. The principle states that these tasks should be separated in architectural QoS frameworks. One aspect of this separation is tile distinction between signaling and media-transfer. Flows which are isochronous in nature generally require a wide variety of high bandwidth, low latency, non-assured services with some form of jitter correction. On the other hand, signaling which is full duplex and asynchronous in nature generally requires low bandwidth, assured-type services; transparency principle states that applications should he shielded from the complexity of underlying QoS specification and QoS management. An important aspect of transparency is the QoS-based API at which desired QoS levels are stated.

Implementing a POC implies that the partial order and the partial reliability must be known by the communicating entities implementing the protocol. In order to deliver the SDUs to its user in an acceptable order, the receiving entity must know the required partial order and reliability. In case of a missing SDU, it may have to ask for retransmission and store the SDU for postponed delivery in case of sequential delivery.

In order to establish the adequate POC, it is proposed that the application passed the Petri net model of the electronic commerce object a PO and a partial reliability to the sending entity. The latter will transmit this information, when establishing the partial connection, to the remote receiving entity as a connection parameter.

Upon receiving it, the receiving entity then builds a local representation of the model including the partial order and reliability information, and uses it to accept the SDUs and deliver them to the user.
Of course, the corresponding SDUs are delivered not necessarily in the order in which they have been sent, however always in an order compatible wit the agreed partial order.

3.2. A new electronic commerce transport protocol architecture

Let us now consider and end-to-end architectural view of the POC by relating it to the ISO transport layer.

This section presents an architecture for electronic commerce transport protocols and shows how to design such an architecture using the AAN approach.

It seems now clear that the best end-to-end transport design architectures are the ones that use the underlying network in the most efficient way. As different electronic commerce component have different types of QoS requirements, and effective use of the under lying network means the use of different QoSs, i.e. different connections providing different QoSs.

A electronic commerce objects can be decomposed into different sub-objects, each having a given QoS. The set of all QoSs define a set of media connections. As media connections may take different routes and as network may lead to different delays, the transferred object may be desynchronised. Upon receiving it, the remote entity has to correctly resynchronise and deliver it.

In the architecture, a given connection transfers the SDUs corresponding to a given QoS, and the role of the POC is to menage these media connections and their electronic commerce synchronisation.

In the general case, each media connection contains a media transport protocol, which needs to be high speed for video sequences. The POC handles the error control, the synchronisation and the delivery of the SDUs to the transport users: this POC transport layer is in charge of controlling all transmission errors according to the agreed partial reliability and partial order.

Inside this new transport layer, shows the untimed Petri net model which is used to control the electronic commerce transfer.

Let us now show why partial reliability and order imply the delivery of the SDUs as early as possible.

Let us assume that a given media transport connection *C* ahs a reliability parameter *R*. This parameter can be defined in several ways. Consider here for simplicity that *R* is defined by a simple integer value, the maximum number of SDUs that may be consecutively lost in sequence. This means that if the number of lost SDUs is less or equal to *R* no action is taken; if it is greater than *R*, the protocol has to perform appropriate recovery actions.

Two types of delivery controls exist:

(a) With separate business control. This control involves two sub-layers. In the first sub-layer, each media connection possesses its own transport protocol. Each media protocol, which must be partially reliable, may be implemented using, for instance, and extension of XTP. In this case each media transport protocol handle control and recovery of lost SDUs. The second sub-layer, including the partial order model, i.e. the PO model sub-layer, enforces the order the synchronisation between the different connections. It does not deliver a SDU received on a transport connection if its delivery is forbidden by a SDU of another connection which preceed the latter according to the given partial order and not yet received on this other connection.

(b) Grouped business control. More efficient, it will be considered now and illustrated below by an example.

This type of per group of business control, which leads to earlier delivery, considers the electronic commerce connection as an integrated object having no sublayer. Each media connection is sequential and controlled by a partial order which handles, at the same time, error control and order control. To make the delivery more efficient, the general partial order transport protocol now delivers a given SDU on a connection C_i as soon as possible, by considering all media connections. In particular, it will deliver a SDU even if its delivery generates the loss of one or more SDUs on another connection or on other connections, but only if this loss is acceptable with respect to the QoS of the corresponding connection.

As demonstrated by an example below a given SDU belonging to a given connection C_i will be delivered according to R_i. The reliability of C_i, and may even be delivered if its delivery implies a loss of one several SDUs on another connection C_j, provided that the target reliability R_j is satisfied. In this case a not yet delivered SDU(s) may be forced to be lost if an ordering constraint implies that it cannot be delivered after the first SDU. This permits the early delivery and he early indication of the losses, as in the following example.

The advantage of first indicating the loss what we have implemented will become clear in the next section.

We note that this protocol delivers the SDUs at the earliest dates: as a fundamental consequence, it provides the best delay and jitter values.

Finally, in complex applications, several POC electronic commerce connections may be needed. As before, each POC electronic commerce connection will be defined by a partial order and a partial reliability. Different electronic commerce POCs will have to be instantiated, possibly corresponding to different, independent electronic commerce objects, just like an application may use several TCP connections.

3.3. Agent Manager

The main purpose of an agent manager is to facilitate the establishment of logical structures to support collaborative sessions between agents and servers. Agent managers form a multicast group and communicate on a common information channel to advertise the profiles of their associated servers and assist agents in identifying suitable servers for the execution of a task,

When an agent manager receives an agent advertising the service profile of a server, it first authenticates the agent and its associated server. The agent manager then extracts the server's profile inforfnation in order to determine the use the advertised service based on the profiles of the local agents and the recent execution contexts of workflows.

On the other hand, if the agent manager determines that no match exists between the agent query and currently available service profiles. If the number of selected servers does not meet the minimum number of servers required by the underlying application, the agent initiates a procedure to seek new servers. To achieve this the agent progressively visits the neighboring agent managers, the closest neighbor first and the neighbors of the neighbors, etc. During this trip, the agent itself may carry advertising information about its local servers to neighboring agent managers.

3.3.1 Agent Group Communication Server

Depending on the underlying application, the interaction between agents and servers may require provision for multicast capabilities and efficient managertwnt of multicast group addresses. These groups are short lived and usually last for the duration of the workflow execution. Consequently, an efficient scheme to manage dynamic allocation of multicast addresses in the agent framework is an important design issue that need to be addressed. While the Internet protocol suite provides a number of proposals for multicast protocols, but these do not address the issue of dynamic allocation of Rroup addresses. The main objective of the agent group communication server is the dynamic allocation of multicast addresses in support of group communication among agents, agent managers and servers.

The benefits of transparency are that it reduces the need to embed functionality in the application. The hides the detail of underlying service specification from the application and it delegates the complexity of handling QoS management activities to the underlying framework; multiple timescales principle guides the division of functionality between architectural modules and pertains to the modeling of control and management mechanisms.

A nice consequence of this approach is that the application has no longer to wait for the end of the maximum time interval in case of lost frames. Furthermore, no new error or losses are introduced by the application itself and the application proved to be able to recover from such short delays.

4. Implementation Plan

We are planning on building a prototype of a dynamic workflow system that integrates workflow with mobile agents implementing our notion of agents. As a first step, we plan on using a mobile agent technology developed that secure communication.

An mobile agent applet is a lightweight mobile Java object approximately 2 kilobytes. If the mobile agent m equipped with database capabilities and Java

Database Connectivity (JDBC) Drivers, they become DBMS mobile agent that can connect to a remote data sever and collect data and performed updates. An mobile agent applet carries along its program code, state, unique identification, and query trip plan as it moves from host to host. If there are any communication problems, such m a host failure, the trip plan allows the agent to try alternate hosts and solutions. A host interacts with an agent by utilization agent server program. The java program listens for incoming agents and provides a context in which they can resume their suspended execution. Agents can be created, talk to one another, move from host to host, and be disposed of at any time.

Our preliminary implementation plans consist of developing the agent prototype over agents. incorporating the advertising strategies and dynamic selection of workflow segments and adding efficiency enhancing features such as multicasting. This will prove to be relatively easier since several features of agents such as migration between hosts are already available in the agent technology. Furthermore we can concentrate our efforts on the interactions between the workflow system and agents.

5. Conclusion

This paper presents a architecture for designing distributed electronic commerce systems. The proposed approach is based on the use of a formal model that describes the complex synchronisation requirements of a general electronic commerce object.

We have described a framework to enable the efficient operation of a virtual enterprise (VE) composed of multiple distributed workflows. The major goal of this paper was to develop a framework that enables the automatic establishment of a VE and maximizes the sharing of useful information between the various components of the VE without human intervention as done in previous strategies. The core of our approach is based on mobile agents referred to as agents. Agents are dynamically created on a need to basis and roam a specified relevant area of the workspace to gather new information and facilitate cotinections between related workflow.

Although other models could have been used, it has been shown that Petri net based OCPNs and TStreamPNs were able to nicely express timed synchronisation, to support the partial order connexion (POC) concept, and to develop a new electronic commerce transport architecture, which has been called application-aware networking.

This approach leads to the definition of a partial order communication service provided by a so-called partial order connection. A corresponding protocol can be derived which uses the electronic commerce object model for its control. Current work addresses Hierarchical Time Stream Petri nets to derive new communications architectures more adapted to hypermedia objects. Efficient implementations of high speed electronic commerce POC transport protocols on top of ATM networks are also being developed. Finally, extensions of the present proposal are under investigation in order to include sophisticated negociations and renegociations of the POs.

Our work is related to the work on adaptive, evolving and migrating workflow. In particular migrating workflow share the same underlying idea with our work, treating a static workflow specification as incomplete and attempt to dynamically expand it by migrating the execution of the workflow front one service site to another until it achieves its goal. They also suggest the use of mobile agents as a potential implementation infrastructure.

Currently, we are investigating different negotiation strategies and contract establishment methods as part of the refinement of our framework. The response time and network utilization for an office type environment using the proposed protocol were evaluated and were shown to be significantly better than those obtained by using protocol.

References

[1]Alonso C., D. Agrawal, A. El Abbadi and C. Mohan. “Functionalities and Limitations of Current Workflow Systems.” IFEE Expert, 12(5), 1997.

[2]Baflardie T., R Francis and 1. Crowcroft. “Core Based Tree (CBT): An Architecture for Scalable Interdomain Multicast Routing.” Proc. of the ACM SIGCOMM Symp., pp. 85-95, 1993.

[3]CasaLi F., S. Cori, B. Pernici and G. Pozzi. “Workflow Evolution,” Data & Knowledge Engineering, 24(3)i 211-238,1998-
[4] M. Diaz, G. Pays, "The Ceasme project: Formal design of high speed multimedia cooperative systems", Special issue on CESAME, Annals of Telecommunications, May-June 1994.
[5] P. Merlin, D.J. Farber, "Recoverability of communication protocols", IEEE Trans. Communications 24 (9)(1976).
[6] Keshav, S., (1993),".Report on the Workshop on Quality of Service Issues in High Speed Networks", ACM Computer Communications Review, Vol. 22, No. 1, pop 6-15, January.
[7] Kurose. J.E. (1994) "Open Issues and Challenges in Providing Quality of Service Guarantees in High Speed Networks", ACM Computer Communications Review, Vol. 23, No. 1, pop 16-25, January.
[8] Miloucheva, I.(1995) "Quality of Service Research for Distributed Multimedia Applications". ACM Pacific Workshop on Distributed Multimedia System.
[9] Vugrl. A.. Bochmann. G v,, Disallow, R., Geckos, j. and B Kerherv, (1994) ,"Distributed Multimedia Applications and Quality of Services A Survey". IEEE Multimedia. 1994.
[10] Volg, C., Wolf, L., Herrtwich, R. and H. Wittig, (1996) "HeiRAT - Quality of Services Management for Distributed Multimedia System", Multimedia System Journal.
[11] Zhang, L., et. Al., (1996) "RSVP Functional Specification", Workink Draft, draft-ietf-rsvp-spec-10.ps,.
[12]T. Little, A. Ghafoor, "Network considerations for distributed multimedia objects composition and communication", IEEE Network, November 1990 pp. 32-49.
[13]T. Little, A. Ghafoor, "Synchronization and storage models for multimedia objects", IEEE J. Selected Areas in Comm. 8-3 1990 pp. 413-427.
[14] S. Prabhakran, S.V. Raghavan, "Synchronisation models for multimedia presentation with user participation", in: Proceedings ACM Electronic commerce, 1993.
[15]R. Botafogo. D. Mosse, "The MORENA model for hypermedia authoring and browsing", in: Proceedings of the IEEE International Conf on Multimedia Modeling and Systems, Washington, 1995.
[16] H.K. Wang, J.L.C. Wu, "Interactive Hypermedia Applications: A model and its implementation", Software Practice and Experience 25 (9) (1995).
[17] P. Senac, P. de Saqui Stannes, R. Willrich,. "Hierarchical time stream Petri nets: A model for hypermedia systems", in: G. de Michelis, M. Diaz (Eds), Proceedings of the 16th Internatonal Conf. on Application and Theory of Petri Nets, June 1995, Torino, Lecture Notes in Computer Science, vol. 935 (Springer, Berlin. 1995) pp. 451-470.
[18]P. Senac, M. Diaz, A. leger, P. de Saqui Sannes. "Modeling logical and temporal synchronization in hypermedia systems", IEEE J Selected Areas in Communications 14 (1) (1996) pp 84-103.
[19] J.P. Courtiat, Cruz de Oliveira, "About time non-deteminism and exception handling in a temporal extension of LOTOS, in: Vuong, Chanson (Eds.)", Proceedings of the IFIP International Conference Protocol Specification, Testing and Verification, Vancouver, 1994, North-Holland.
[20]GeorgakGpoulos D., H. Sinha, K. Huff and B- Hurwitz. 'Monitoring Multi-organizational Processes." Proc-of the ll th Int'l Conf on Parallel and Distributed Computing System, pp. 75-80, 1998.
[21]Han Y. and A. Sheth. “On Adaptive Workflow Modeling.” Proc. of the 4th Int'l Conf on Information Systems Analysis and Synthesis, pp. 108-116, 1998
[22] Jablonski S. et al. "External and Internal Support Services in Workflow Management Systems”Proc. of the 11 th Int'l Conf. on Parallel and Distributed Computing Systems, pp. 81-86,1998.
[23]Lange D. and Oshima M. “Programming and Deploying Java Mobile Agents with mobile agent,” Addison Wesley, 1998
[24]Moy J..” Multicast Routing Extensions to OSPF.” Communications of the ACM, Aug. 1994.
[25]Papastavrou S., E. Pitoura, and G. Samaras. “Mobile Agents for WWW Distribued Database Access.” Pmc. of the 15ih Ini'l Conf. an Data Eng., 1999.