The Sixteenth International Symposium on Artificial Life and Robotics 2011 (AROB 16th '11),
B-Con Plaza, Beppu,Oita, Japan, January 27-29, 2011

Efficient Distributed Ontology Management Scheme for Inference in Surveillance Networks

S. Yang

Department of Information Engineering,
The University of Suwon, 445-743, KOREA
(Tel : 82-31-229-8138; Fax : 82-31-229-8284)
(Smyang@suwon.ac.kr)

Abstract: In this paper, we propose an efficient and scalable surveillance network providing better context inference based on distributed ontology framework. Upon our distributed ontology management system, each agent can build and process ontology cooperatively. They share context ontology for cooperative combined inference. Data source servers not only can get services from a region server, but also they can form and generate a P2P(peer-to-peer) network to provide services to each other. For an efficient and scalable cooperation we adopt a P2P communication through common API and adaptive caching.

Keywords: Surveillance network, Agent, Context ontology, Peer to peer network**.**

I. INTRODUCTION

In our network surveillance environment, each surveillance devices is equipped with a customized agent. The distribution of demands for multimedia data items is often skewed, and the surveillance devices have different capabilities and data formats. These can lead to poor data communication and dropped messages resulting in narrow reasoning and decision. Therefore for more easy and efficient cooperation between the agents, we propose a more efficient and scalable communication framework utilizing P2P computing and an adaptive cache scheme based on heterogeneous devices and data network.

Context is any information that can be used to characterize the situation of environment entities. To overcome the variety of data source servers, we build agent for each and interface through common APIs. Our network surveillance application can be thought of semantic web. Each agent tries to be context-aware through integration, analysis and inference of data and information. For better inference, they need not only data of their own but also data of other neighbor agents. P2P computing utilizing common API provides flexible, scalable and efficient ontology sharing method.

In this paper, we also propose a distributed adaptive cache scheme based on heterogeneous device and data network. Our scheme uses caching and conformity to update and share data in a cooperative way. Experiments and implementation are conducted to evaluate the effectiveness of our flexible cache scheme.

The rest of the paper is organized as follows, Section 2 we suggest a distributed inference framework for surveillance networks. Experiments and implementation is explained in Section 3. Section 4 reviews the related work on distributed inference based on ontology, particularly focusing on the user interface and the adaptive caching technique. Section 5 concludes this paper with a short summary.

II. DISTRIBUTED INFERENCE FRAMEWORK

Our surveillance network constructs a web of inference systems. Each data source server is equipped with an independent agent having local context ontology and inference engine. Cooperative inference such as object tracking requires the cooperation among local data source servers. Data associations between data sources are required to cover larger areas and solve occlusion problems. Our distributed inference framework manages ontology processing, ontology location and ontology connection on behalf of each agent.

Data source servers form a graph structure. For the high level inference, regional hierarchies are included in it. Each data source server can communicate each other freely within access control permission forming P2P computing networks. Equipped with agents, they perform their own intelligent distributed inference based on their own ontology knowledge base. In the case of multi-agent systems, association of data across the data source servers is indispensable for context awareness and decision making for the integration of ontologies and prevention of the narrow decision.

For each data request, data source server n_i first tries to find the required data item from its local cache. If it encounters a local cache miss, it broadcasts a request message to its neighbor peers via P2P broadcast communication. Our ontology framework includes common APIs for the operations. If a neighbor peer does not have the needed data item, the agent will send the request to the higher region server instead to obtain the needed data. The representative common APIs are getVideoSource(), getVideoSourceConfiguration(), getVideo(), setVideo(), getVideoStatistics(), getVideoCaptureSize(), getUsers(), getVideoResolution(), requestSignal(), sendSignal(), getBitrateControlMode(), etc.

For the broadcasting, the latency is defined as the sum of the request and reply packets divided by network bandwidth through the hops the data gets transmitted. As peers can exchange such information as the cache sizes of their neighbors and the transmission ranges, they can negotiate and come up with a cache strategy based on the messages exchanged. As a result, better usage of their limited cache space and higher system performance can be obtained

The multimedia data and biometric feature information caching will be distributed over the region servers and data source servers. The data source servers need to carry out the indexing and retrieval of the information distributed across the servers in an efficient manner. To maintain the freshness and effectiveness of the data, we should cache data adaptively. We measure the weight of cached data to describe its relative importance as compared to the other data as proposed in [4]. The higher the weight, the lower is the probability of the data being replaced. We also use a policy based on the size of the objects, in which the weight is proportional to the size of the data. Therefore the weight is computed as the product of the access frequency, size of the data and the data recency:

$$Weight = Frequency^{f}\, DataSize^{s}\, \mathrm{Re}cency^{r}$$

The exponents are weighting factors. As a result, we can expect the packet traffic would be proportional to the probability of cache miss, latency, the probability of packet loss, the weight of data and the inverse of the number of hops.

III. EXPERIMENTS AND IMPLEMENTATION

In a distributed surveillance environment, multimedia data and biometric feature information are produced continuously. Furthermore the multimedia data transmission consumes large network bandwidth. When the adaptive caching technique is applied, experiments show better performance as shown in Figure 1. The data source servers might be grouped by region. The caching skewed by region could be reflected through the variance in caching ratio.

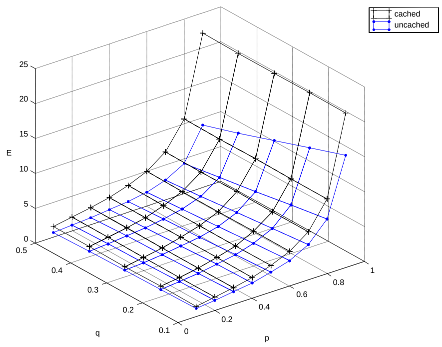

Fig.1. The expected packet transmission comparison

For the integration of ontologies between data source servers, we developed common APIs according to the standard [7]. The pseudo code for the representative common APIs is as the following:

```
function grrc(key){
    this.name = 'grrc';
    this.key = key;
    this.XmlHttpObject = function(){ //abridged
    }
    this.validChk = function(){
        var add;
        add = location.href;
        add = add.split("//");
        add = "http://" +
        add[1].substr(0,add[1].indexOf("/")) + "/";
        host = 'http://localhost/grrc/';
        URL = host + 'validChk.php?key=' + this.key
        + '&add=' + add;
        xmlhttpr =   new XMLHttpRequest();
        xmlhttpr.open('GET', URL, true);
        xmlhttpr.send(null);
```

```
        xmlhttpr.onreadystatechange ();//abridged
    }
    this.GetUsers = function(){
        this.validChk();
        host = 'http://localhost/grrc/';
        URL = host + 'getUsers.php';
        location.href = URL;
    }
    this.GetVideoSource = function(){
        this.validChk();
        host = 'http://localhost/grrc/';
        URL = host + 'getVideoList.php';
        location.href = URL;
    }
    this.GetVideoSourceConfiguration =
        function(){
        this.validChk();
        host = 'http://localhost/grrc/';
        URL = host + 'getAllVideoList.php';
        location.href = URL;
    }
    this.GetVideoCaptureSize = function(){
        CaptureSize = new Array(1);
        CaptureSize[0] = new Array(2);
        this.validChk();
        host = 'http://localhost/grrc/';
        URL = host + 'VideoInfo.php';
        xmlhttp = this.XmlHttpObject();
        xmlhttp.open('GET', URL, true);
        xmlhttp.send(null);
        xmlhttp.onreadystatechange = function();
        //abridged
        return CaptureSize[arrWidth][arrHeight];
    }
    this.GetVideoResolution = function(){
        Resolution = new Array(1);
        Resolution[0] = new Array(2);
        this.validChk();
        host = 'http://localhost/grrc/';
        URL = host + 'VideoInfoResolution.php';
        xmlhttp = this.XmlHttpObject();
        xmlhttp.open('GET', URL, true);
        xmlhttp.send(null);
        xmlhttp.onreadystatechange =
        function();//abridged
        return Resolution[arrResolutionWidth]
        [arrResolutionHeight];
    }
    this.getVideo = function(num){
        this.validChk();
        host = 'http://localhost/grrc/';
        URL = host + 'getVideo.php';
        xmlhttp = this.XmlHttpObject();
        xmlhttp.open('GET', URL, true);
        xmlhttp.send(null);
        xmlhttp.onreadystatechange =
        function();//abridged
    }
}
```

Our work is aimed at designing and deploying a system for the surveillance and monitoring of province area containing about 15 cities. Final product is to be included in the construction of the ubiquitous city. The agent platform should be distributed across various data acquisition systems such as CCTV, RFID sensors, sound monitors, etc. We show the live video and the stored video to monitoring clients and devices are controlled via a remote GUI even from a smart phone. Figure 2 shows screenshot of map interface. It shows the stored event data. On the left frame we can see the google map. On the right frame, alarming events are listed first with features. When we select the CCTV, we can see the specific videos on the bottom side.

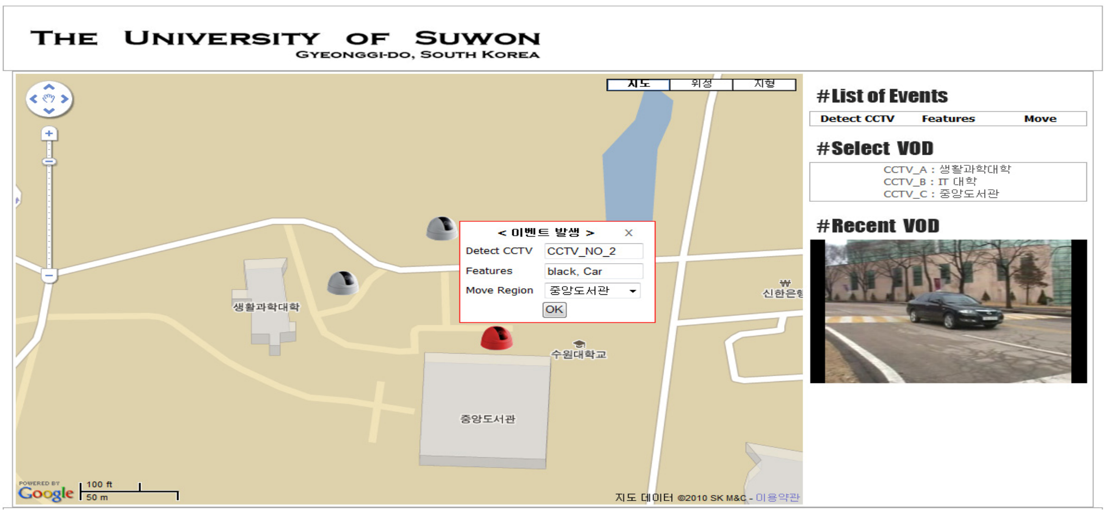

Fig.2. The sample user interface with google map

IV. RELATED WORK

Several ontology reasoning systems have been developed for reasoning and querying the semantic web and they show good performance as seen in [1,2,3]. We want to adopt, customize and improve them adequate for our surveillance system. Especially all of them do not use common API for communication. In [4], a cooperative caching framework is introduced and claimed to be effective to data availability. In [5], a replica allocation method and clustering in distributed networks are introduced to improve data accessibility in a mobile communication environment. Another cooperative cache scheme for similar peers is described in [6], which combines the P2P communication technology with a conventional mobile system. However they do not apply their technique to the distributed surveillance system.

V. CONCLUSION

We describe an efficient ontology management scheme for better context inference based on distributed ontology framework. Data acquisition servers communicate each other freely within access control permission to perform their own intelligent distributed inference based on integrated ontology knowledge base utilizing common APIs. Our scheme uses P2P computing using common APIs and data conformity to update and share data in a cooperative way. For such a P2P network, an effective cache framework that can handle heterogeneous devices is required. We also present a flexible cache scheme which is adaptive to the actual device demands and that of its neighbors. We analyzed our distributed ontology management scheme through experiments and implementation.

VI. ACKNOWLEDGMENTS

This work was supported by the GRRC program of Gyeonggi province. [GGA0801-45700, Center for U-city Security and Surveillance Technology]

REFERENCES

[1] A. Flahive, W. Rahayu and D. Taniar, (2005), A Distributed Ontology Framework in the Semantic Grid Environment, 19th International Conference on Advanced Information Networking and Applications (AINA'05)

[2] J. Z. Pan, (2007), A Flexible Ontology Reasoning Architecture for the Semantic Web, IEEE Transactions on Knowledge and Data Engineering archive Volume 19 , Issue 2, 246-260

[3] Anne Schlicht, Heiner Stuckenschmidt, (2008), Towards Distributed Ontology Reasoning for the Web, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

[4] L.Yin, G. Cao, (2006), Supporting Cooperative Caching in Ad Hoc Networks. IEEE Tr. On mobile computing, Vol. 5, No. 1.

[5] Zheng, J., Su, J., and Lu, X. (2005), A Clustering based Data Replication Algorithm in Mobile Ad Hoc Networks for Improving Data Availability, Lecture Notes in Computer Science 3358

[6] Chow, C., Leong, H., and Chan, A. (2005), Distributed Group-based Cooperative Caching in a Mobile Broadcast Environment, MDM 2005

[7] ONVIF, 2009, Open Network Video Interface Forum Core Specification version 1.01