JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIES
JULY 2011 VOLUME 5 NUMBER 2 (13-24)

AN ARCHITECTURAL MODEL FOR CONTENT MANAGEMENT IN E-COMMERCE APPLICATIONS USING INTELLIGENT AGENTS

Akhan AKBULUT[*]

Istanbul Kültür University
Computer Engineering Department
Ataköy-İSTANBUL
a.akbulut@iku.edu.tr

Güray YILMAZ

Turkish Air Force Academy
Computer Engineering Department
Yeşilyurt-İSTANBUL
g.yilmaz@hho.edu.tr

Received: *01st May 2011,* ***Accepted:*** *27th July 2011*

ABSTRACT
In e-Commerce applications, the size of the architecture is huge and there are too many member shops and too many customers that bring along management difficulties. So, different technologies need to be used and usually autonomous structures work with unmanned decision-making processes for the managerial processes of an e-Mall where hundreds of e-shop catalogues are displayed. In this study, our recent works are developed and their applications are actualized. Additionally, an architectural solution that will ease the content management for an e-Mall structure is offered.

Keywords: *Intelligent agents, e-Commerce, e-Malls, content management, clustering, hashing.*

E-TİCARET UYGULAMALARINDAKİ İÇERİK YÖNETİMİ İÇİN AKILLI AJANLAR KULLANAN BİR MİMARİ MODEL

ÖZET
e-Ticaret uygulamalarında, mimarinin çok büyümesi, çok fazla üye mağaza ve müşteri olması yönetimsel zorluklar ortaya çıkarır. Yüzlerce üye mağaza kataloglarının sergilendiği bir e-Alışveriş Merkezindeki yönetimsel işlemler için insana ihtiyaç duymadan çalışan karar destek sistemleri gibi farklı teknolojilerin kullanılması gerekmektedir. Bu çalışmada, önceki araştırmalarımız geliştirilmiş ve uygulamaları gerçeklenmiştir. Ek olarak, bir e-Alışveriş Merkezi yapısı için kullanılanbilecek içerik yönetim sistemi mimarisi sunulmaktadır.

Anahtar Kelimeler : *Akıllı ajanlar, e-Ticaret, e-Alışveriş Merkezi, içerik yönetimi, kümeleme, anahtarlama.*

1. INTRODUCTION

E-commerce and E-business are the major contributors to the current emergence of Digital Economy [3]. Today, e-commerce has some management difficulties that users are not aware of. Product management is not a problem for singular online stores. However, when the store numbers reach hundreds for one site, much more complicated decision-making mechanisms must be used. Rather than some big e-Malls, there is no suitable management infrastructure for the product structure. In this paper, we propose a method that can be applied to any e-commerce site, before or at the time of installation when it is determined that the data consistency of the product management should be increased.

At the big e-commerce and comparison sites like Kelkoo [4], Shopzilla [5] or Froogle [6], the system structure is set upon a central management and consists of the services given by this central structure. In general, these services are supplied by web services that answer the authentication, product management and reporting needs of the member stores. The most critical issue for these services is product management, because it causes many more problems.

Member stores update their products by uploading their catalogues in the systems required format. Since there is no standard format for products' feeds, XML

[*] Corresponding Author

is generally preferred because of its performance criterion. But, there are also various infrastructures available that use a tab-delimited text-based format like Froogle.

Whatever format is used, the information sent includes product names, detailed explanations, prices, product images and their categories. Member stores use this information in their structures and also check them in a centralized way. For example, a member store can exhibit a laptop computer on its site under the heading of "home electronics". By integrating this product into the system with the information at the member store, the same product will be shown in many places with different headings such as "computers", "notebooks". For this reason, data integrity will fail in the system. Since no product category of the member stores will be the same as the others, it will become harder to fit them into the system. At this point, a smart decision-making mechanism that will decide in which category to include the product must be used. We advise to assign this duty to our proposed AGent Based E-commerce Platform (AGBEP) [1, 2].

2. PRODUCT CATEGORY IDENTIFICATION PROBLEM

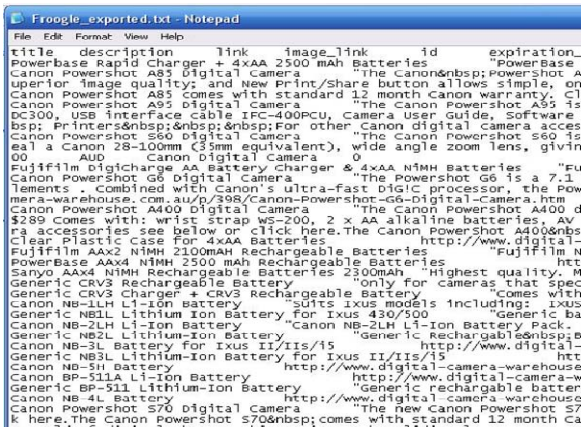
The systems embodying the information that the member shops send appears to be the biggest problem. Because of the possibility that each member shop may send inaccurate information about the products with their categorization system, they need to be checked by an autonomous and proactive intelligent mechanism.

The product portfolio of e-commerce MALL (e-Mall) sites has included member stores. Various technologies have been used while transferring product information from member stores to the e-Mall. The information is mostly transferred either by using web services or different remoting infrastructure in XML format (Fig. 1.a) [7]. The primary goal of XML is to provide a marking text component and use such data for exchange among information sources [8]. Text based feeds (Fig. 1.b) have also been used instead of XML.

Whatever type of format has been chosen for the feed, the given information it contains is exactly the same. It is important that the feed contains the information of each product in itself and that the e-Mall database is updated with this information.

There needs to be a mechanism which will be able to find out the product names of these feeds that are sent in the system and match them. If there exists an undefined product in the system, with the help of the product explanation information in the feed, we will be able to find out what category to match the product to. We must not trust the category information that the e-shop sends the feed and the system must prefer the category proving method so that there are no replication records, and data integrity is provided in the database.

a. XML-based product feed example for PriceRunner.



b. Tab delimited text based product feed example for Froogle.

Fig. 1. Technologies used to transfer product info from member stores to e-Malls.

2.1. Content Classification Techniques

Content classification is categorization technique for document datasets given categories in advance. Another definition can be expressed as; process of assigning documents to one or more predefined categories based on their content. This technique has been a well studied problem and it is used to solve in many scenarios such as classifying news [9,10], classifying and archiving emails [11], classifying web pages [12] and categorizing stories [13]. All these applications build on several approaches. Some of the approaches rely on setting up topic definitions that require selection of related keywords or phrases. This method is consisting of several tasks which are works to extract relational information from text. Additionally, some statistical approaches are applied to solve this problem. Vector Space Model (VSM) is used in this type of approach with good classification accuracy. This method based on the k nearest vector technique. Content classification also been tested with neural network architectures.

To analyze the product category identification problem, clustering and hashing methods are used. To identify the product's category, we must find out how close are the sent keywords to the keywords used in the products' explanations of the defined products by grouping them. The preferred method for the process of dividing them into groups is clustering. Our method is the combination of the two approaches which are explained above.

The preferred clustering methods inside AGBEP are Fuzzy C-Means and K-Means clustering algorithms. But, we will not be able to use the explanations that come from feed as it is sent. Because the words from the explanations are alphanumeric, the clustering process becomes much more difficult. In order to prevent this difficulty and turn the chosen keywords to numeric values, the hashing methods are used.

3. AGBEP: AGENT BASED E-COMMERCE PLATFORM

As it was mentioned in Section Two, adding the information in catalogues which the member shops sent to the e-Mall by using the agent technologies, and to offer the most suitable infrastructure for an e-Mall are the basic duties of AGBEP. The member shops and the customers roles inside the system needs are planned to yield the highest availability and the best performance. In order to provide these services, servers are configured in different ways for different purposes.

As illustrated in Fig. 2, the structure of AGBEP is directed by a front balancer at the front end in order to resist overload. Visitors are transferred to an available and loosely-loaded web server. Web servers connect to the databases on the SQL 2008 servers by their load balancer background in order to answer the needs of the users. Furthermore, there exists an Agent Server that enables the managerial structure and keeps the agents above.

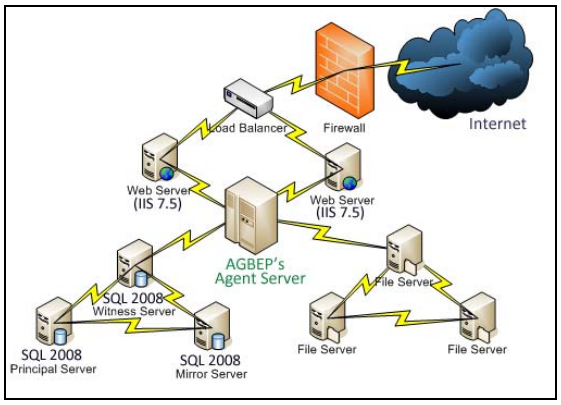


Fig. 2. General overview of AGBEP.

In order to analyze the physical structure in detail, we have to observe it in four distinct parts according to duties and working styles. The first part is the web servers that contain the AGBEP's web interfaces and the server that makes the load balancing of these servers. The Internet Information Server (IIS) 7.5 service of the Windows Server 2008 is used for the hosting of the web pages. The Network Load Balancing, which is one of the built-in services of Server 2008, is also used for load balancing process.

The second part is a platform named Agent Server which enables the agents to do their duties and work. The Agent Server where the Windows Server 2008 operating system is installed contains the JADE framework.

The parts beyond the physical structure are database servers and file servers. The database servers are the computers that use the operating system Windows Server 2008 and contain SQL Server 2008 Enterpriser Edition which Microsoft offers as the most suitable product for the Database Management Systems (DBMS).
The last part consists of the file servers that contain documents with contents such as feeds, product images that the member shops sent.

3.1. Intelligent Agents in AGBEP

The AGBEP's Agent Server contains the agents that are defined to work in different duties and the JADE framework which is an infrastructure [14] that makes these agents work. The agents that work on this server finishes the basic duties that have to be done in an autonomous way according to the duties' contents by sending messages and accessing the servers among themselves.

The first entity of AGBEP is the *Site Administration Agent* (SAA) that is automatically appointed to the member stores by the system. SAA makes the managerial functions continue either by making a data transfer with the other agents or using the system's web services. However, the *Feed Delivery Agent* (FDA) includes the feeds to the system to be analyzed that keeps the product information that the member store sends. An *Inspector Agent* (IA) finds out the keywords by looking at the product descriptions at the feeds that are included in the system. A *Hashing Agent* (HA) then transfers the identified keywords through a hash function and registers to the system. Finally, *Clustering Agent* (CA) decides which product fits in which category within the system by performing a clustering study using these hashed values. Fig. 3 illustrates the whole agent framework and its members.

3.1.1. Site Administration Agent

As a member of the e-Mall structure, Site Administration Agent is used to help the managerial functions of the appointed store. This is the starting point of the feed sending part of the managerial duties of the member store. It takes the feed that the store

submits from the client computer to the FDA and takes back a report of the operation as illustrated in Fig. 4.

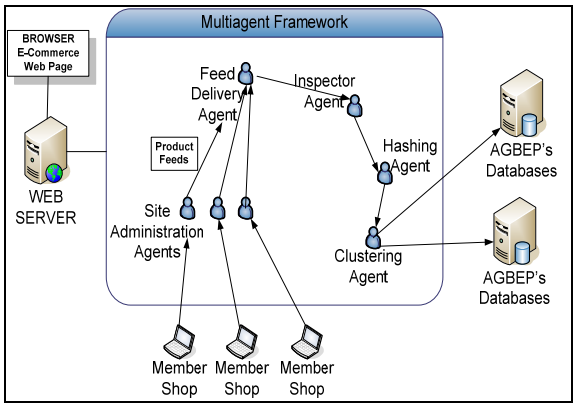


Fig. 3. Agent Architecture in AGBEP.

SSA gives us a safe entry by checking the member shops' user name and password against e-Mall records. It starts the logged e-shop users updating and deleting functions of their product catalogue in the e-Mall. It saves the XML feed over a file server which the user sends that wants to load a new catalogue and warns the Feed Delivery Agent to make the mistake and damage checks. By bringing the success or failure reports to the user which comes back from FDA, it provides a safe logout from the system.

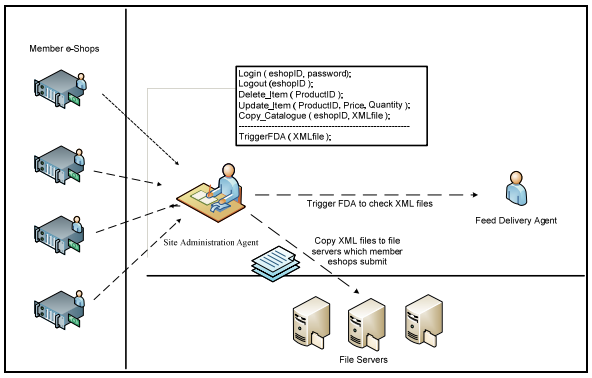


Fig. 4. Site Administration Agent in Process.

We may consider SAA as the starting point of the autonomous categorizing system of AGBEP which is offered as a solution. Beyond submitting the most basic managerial duties, its most critical duty is to unite the XML document which the member shop sends with the member shop ID with the working time and by renaming it. It records them in what is allocated for the member shop over file servers. The information of the XML document which is recorded over file server exceeds the waiting status by transmitting it to the FDA with *TriggerFDA()* method. According to the information coming from the checking process done by FDA, it gives notice of whether the catalogue sent to member shops is taken to action or whether there occurs a mistake.

3.1.2. Feed Delivery Agent

Feed Delivery Agent checks the feeds taken from SAA to the system. In the present system, the feeds are preferred in XML format. It has to check these files in case they get damaged at the time of transfer until the mistake check is done at the XML tags.

After FDA checks the document to see whether it is accessible, it checks the opening and closing tags of all the records in the XML file and it informs SAA.

First of all, FDA checks whether the XML document sent from the member shop is corrupted or not during copying. During this check, it sets to work with the *Check Accessibility()* method. This method presents the result in a boolean way whether it is successful or not by trying to access all nodes inside XML document. After it is understood that there is no problem in accessibility physically, it checks FDA, XML document by *CheckTags()* method whether it is in a format that AGBEP can use meaningfully. During this method, it uses the XSL document defined inside the system.
It takes the elements and attributes in the '*template.xsl*' document as a basis. It checks all the nods in the sent Feed and it sends the result as a boolean value. By checking the results of both *Check Accessibility()* and *Check Tags()*, it decides whether the categorization process will be continued or not, and it sets to work with the *ReportToSAA()* and *TriggerIA()* methods. If the result is affirmative, it activates the next processes with *TriggerIA()* method. The whole process is illustrated in Fig. 5.

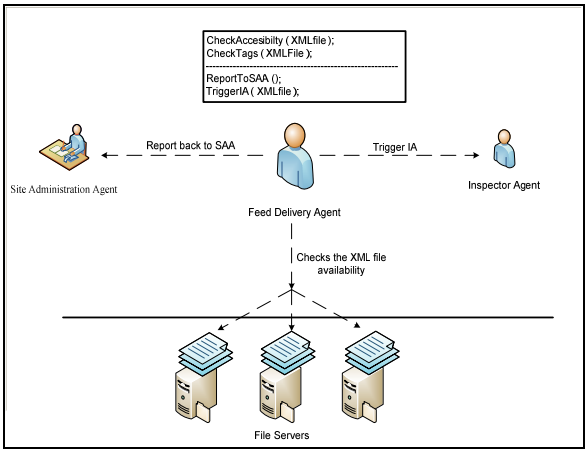


Fig. 5. Feed Delivery Agent in Process.

3.1.3. Inspector Agent

Inspector Agent chooses a certain quantity of keywords from each product description from the incoming feeds, and by grouping the products according to the IDs at the system it delivers them to the Hashing Agent. It is important that the IA work properly while the product category is identified. At the time of identifying the chosen keywords, what criteria are to be used and how many pieces are to be

chosen from these keywords [15] are the most important subject factors in corroborating the categorical operation.

In the process of keyword determination, a random working strategy is chosen. In the starting step of the operation (Fig. 6), all the words that exist in the product information are placed in an ArrayList as they make up separate records. After these records take form, a filtering operation is performed to purge the words that will not be keywords. During the filtration, the records of the words that frequently exist in most sentences such as "the, any, of, that, etc." are erased from the ArrayList.

With the remaining records, a selection process is started by giving priority to numerical statements. During the selection, in order to increase the correctness ratio of category to product matching, we must be careful to choose at least half of the records from non-numerical ones. Otherwise, it could be impossible for the product category to be decided. If half of the keywords are not chosen from the numeric fields or there is not enough numerical fields exist, non-numerical fields will be preferred for selections.

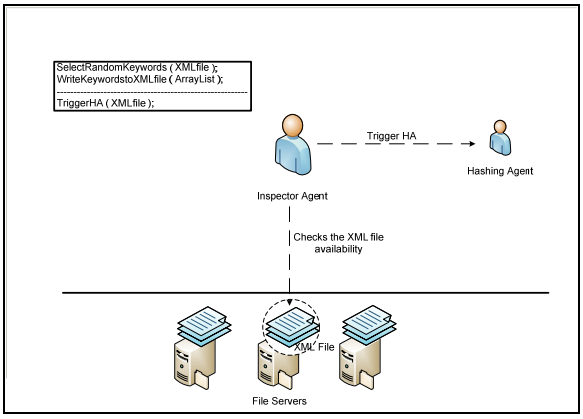


Fig. 6. Inspector Agent in Process.

In selecting the non-numerical keywords, priority will be given to words that come after the measurement units. For this process, the index numbers such as "*cm, pixel, gallon, inch, Hz, rpm, etc.*" of the records' measurement units will be put into another ArrayList. One-third (1/3) of the keywords chosen with a turning rotation (for example, 10 records for a search of 30 keywords) will be filled with words that come after these measurement units. The measurement for this process is to record the index number in ArrayList by increasing the index value by 1. As a result of this process, we reach the words that have importance in deciding the category such as "55 gallon aquarium tank". In determining the other keywords, the remaining non-numerical records are used. Without taking into consideration any priority, the selection is performed at random. Pseudo-code of the keyword selection process is as follows:

```
Open XML file
Read string from Description Tag
Repeat
      Copy each word into Description
      Arrray
Until all words passed
Close XML file
Set wNumber integer by counting words
Set numKeyword by wNumber / 2
For each word in Description Array
      for int i=0 to numKeyword
      if is numeric(DescriptionArray
            [random( )] is true
         write into ArrKeyword
      end
      while i = wNumber
end
Create XML file
Copy keywords to XML file
```

The keywords identified at the end of the whole identification process are transferred into an XML file and are ready for the hashing process.

3.1.4. Hashing Agent

As shown in Fig. 7, Hashing Agent takes the keywords determined by IA feeds from the hashing process and registers them in the database without the attending category. HA converts the keyword records into hashed records using *MD5, SHA1, SHA256, SHA384* and *SHA512* hashing algorithms [16] in order to match selected performance issues in the XML file. After the hashing process, the XML file now contains the hashed records to be clustered.

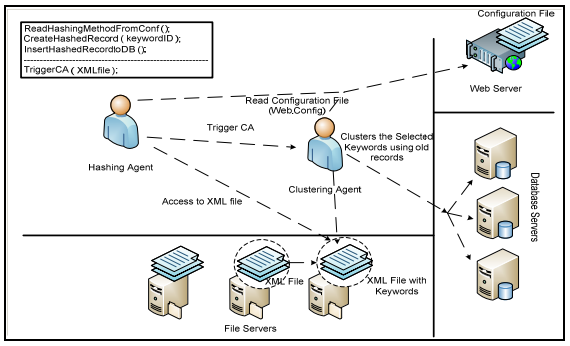


Fig. 7. Hashing Agent and Clustering Agent in Process.

HA puts the keywords from the product feeds' product explanations into hash functions in order to use them in the clustering process and saves them in the e-Mall database. HA reads the hash function it will use in the configuration file of the e-Mall system and applies it. The e-Mall administrators must determine the hashing algorithm that will be used according to the load and overpopulation of the system and must define it as a key to the systems configuration file. Otherwise, HA will use MD5 algorithm as default selection.

3.1.5. Clustering Agent

Clustering Agent checks the products not belonging to a category that are newly registered to the system and determines in which category to include them. By using the present reports at the system, it makes one point for each product at the multi-dimensional space function. It determines where these points come together, which are different categories. If the points are too near each other and come together under a specific threshold, these points can be accepted as the same product and thus are accepted as wrong product descriptions in the member stores. Determination of the boundaries of the categories that are near each other is another important criterion. During the clustering process, *Partition Co-efficiency*, *Classification Entropy*, *Partition Index* and *Separation Index* values must be evaluated correctly.

CA follows up a working method with two levels during the clustering processes using *K-Mean* [17] and *Fuzzy C-Means* [18] algorithms. First, it clusters the product using the keywords in the product information with 100 products that are randomly chosen from each category without taking the threshold value into consideration. It finds out to which category it belongs from among 30 categories extracted from eBay [19]. After the product's category is found, second clustering process is then applied with all database records that belong to that category while keeping the threshold value at a very low level because of its sensitivity.

After the clustering process, it is established to which sub-category (for example, there are sub-categories like Brass, Guitar, Electronic, Equipment under the main category of Musical Instruments) it belongs. When there are products at the limits against the threshold value, which is kept high during the clustering process, they are added to the 'Everything Else' category.

3.2. Configuring Web Servers for Network Load Balancing

AGBEP's web applications typically perform such mission-critical tasks as displaying products, organizing categories, and financial transactions. Businesses can lose millions of dollars when mission-critical applications are not available, so it is important that these applications are available at all times. Briefly, these applications must be highly available and reliable.

An easy and effective way to achieve availability and reliability is to use redundant servers. If you have more than one server hosting your site, when one of the servers crashes, the processing requests can be redirected to another server. This provides a highly available Web site.
To activate the Network Load Balancing (NLB) [20] feature, we have to use the Network Load Balancing Manager, (Fig. 8) which is an administrative console of Server 2008. NLB facilitates the process of creating a Web Server Farm. A Web Server farm is a redundant cluster of several Web servers serving a single (virtual) IP address.

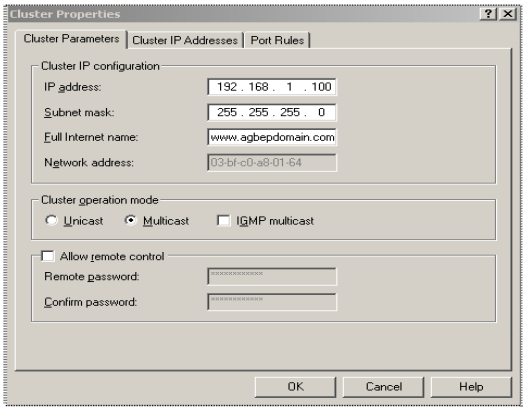


Fig. 8. Properties of a Cluster.

After defining two clusters in the system, the web farm now has two web servers running under a virtual IP to balance the client's request as illustrated in Fig. 9. To Access e-Mall from the web that works under www.agbepdomain.com domain is provided by this virtual IP. With this IP, users that request to enter the e-Mall from their browsers primarily reach the load balancer server. The load balancing service specifies the available cluster in the Web Farm and transmits the incoming request to the chosen cluster.

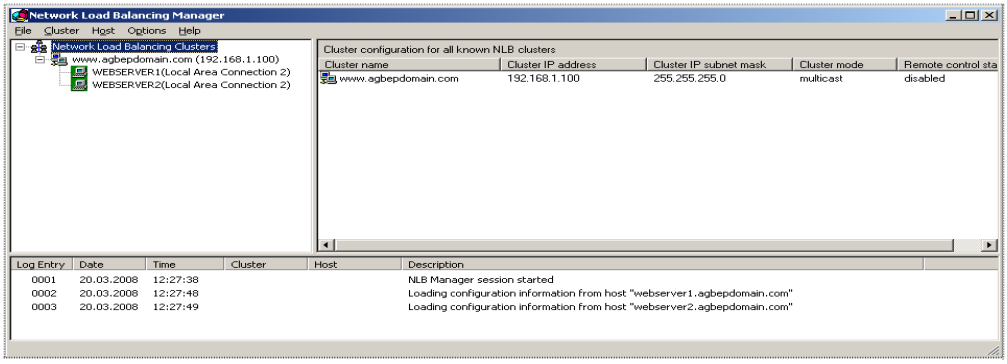


Fig. 9. Network Load Balancing Manager with defined Clusters.

3.3 File Servers Installation and Management

It is very important that all file servers have to be a member of AGBEP domain if we want to control and limit the usage of the member shops as domain users. After including the servers in the agbepdomain.com, we have to install the File Server Management and File Server Resource Manager services.

After setting up the system, we can command the file servers using the File Server Management Console as illustrated in Fig. 10. By using this console, we can create quotas for files which will be uploaded by member shops. Also system administrators can monitor the shared folders and can create scheduled reports of these.

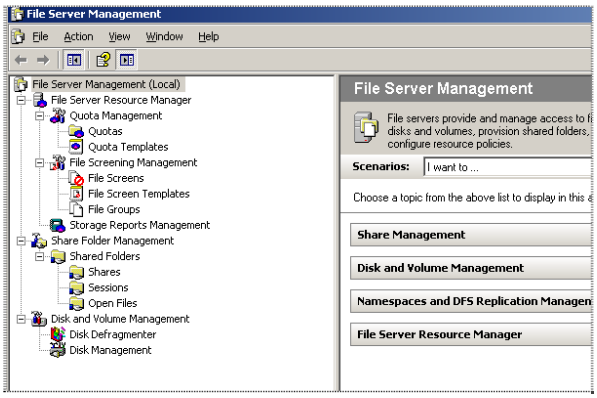


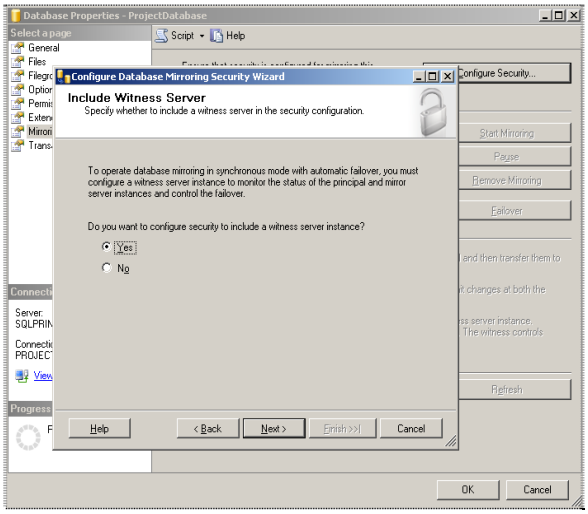
Fig. 10. File Server Management Console.

3.4. SQL Server Database Mirroring Installation

About the problem of maintaining high availability, just because the web server's numbers are enough doesn't mean that AGBEP will work with a high performance. When the web servers are on duty, the back and front database servers operate in order to provide the necessary information. In order to function with a large number of web servers, a large number of database servers need to function similarly. The management of a large number of database servers is another problem that must be solved. Since AGBEP uses the SQL Server 2008 [15] as a database solution, the working methodology of the database servers is managed by Witness Server as shown in Fig. 11.

The witness server is a third instance of SQL Server 2008 that acts as an intermediary between the principal and the mirror in order to determine when to fail over. By having a third instance, it creates the ability to have a 2 to1 vote that says "one of my components is not available and therefore I am going to fail over". Because of the need to determine if the components are online or offline before an automatic failover, the witness server is only needed when you implement the High Availability mode, and when you want or need an automatic failover. This instance does not do much more than communicate with the principal and the mirror to make sure they are still alive. No database activity occurs on this instance, just communication between the three components.

Principal server and the mirror server communicate on a defined port in the configuration as illustrated in Fig. 12.



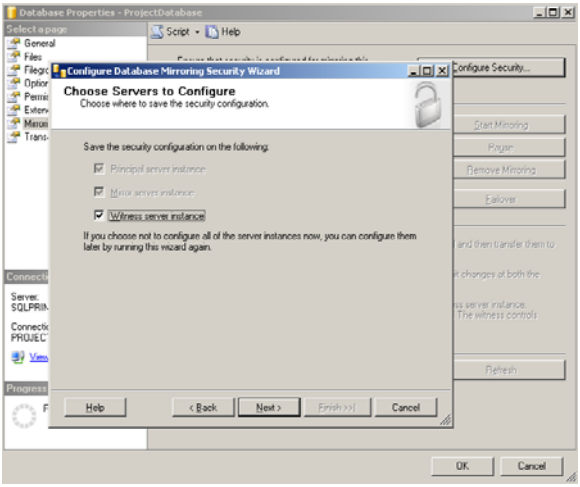


Fig. 11. Installation of Witness Server.

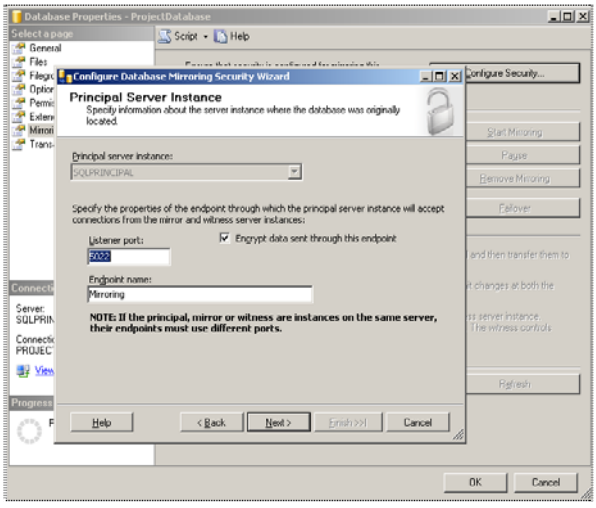


Fig. 12. Setting the Communication Port.

After setting the "Listener Port" and the "Endpoint Name" parameters for all three SQL Servers, configuration will be finished (Fig. 13).

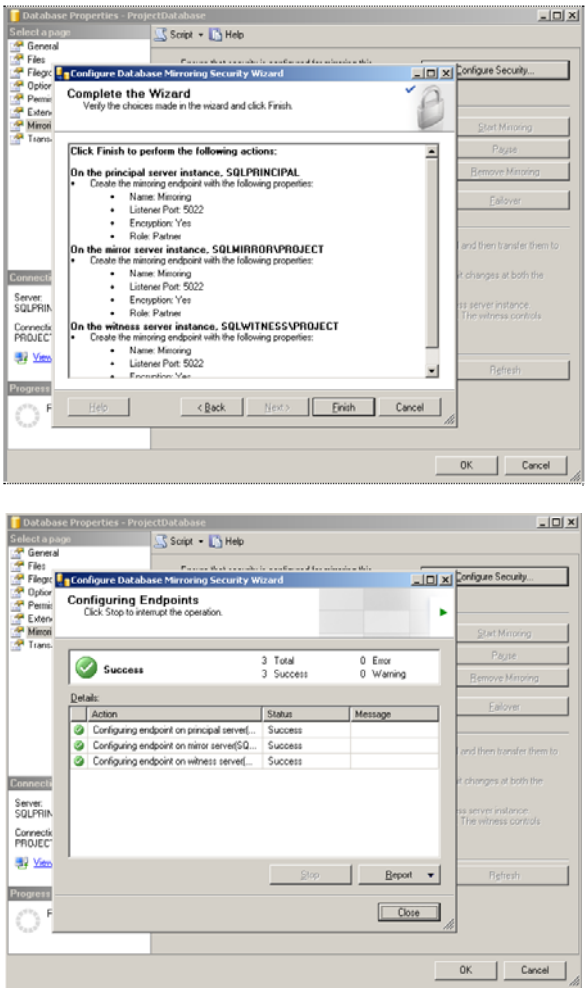


Fig. 13. Confirmation of the parameters.

At the end, we have to install the certificates for authentication of servers before starting the mirroring service as shown on Fig. 14.

Principal SQL Server's certificate can be demonstrated as:

```
create master key encryption by password
= 'AGBEPp@ssw0rd';

create certificate Principal_Server_Cert
with subject = Principal_Server
certificate', start_date = '2008/11/01',
expiry_date = '2030/11/01';

Create endpoint endpoint_mirroring state
= started
as tcp(listener_port = 7024, listener_ip
= all)
for database_mirroring (authentication =
certificate Principal_Server_Cert,
encryption = disabled, role = all);
```

```
Backup certificate Principal_Server_Cert
to file = 'c:\
Principal_Server_Cert.cer';
GO
```

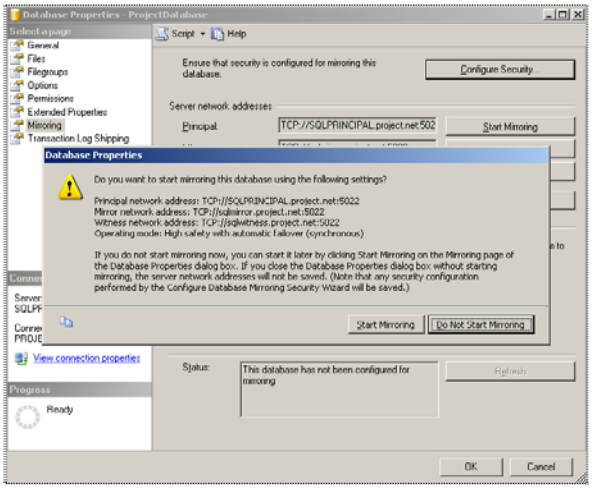


Fig. 14. Starting up the Mirroring Service.

3.5. Composition of the Database

AGBEP has to contain a database with a suitable architecture in order to store the critical information such as products, users, categories, member shops, sales percentages and stock conditions. This database is used by giving access to logins that have access rights to different tables. The table indexes are updated with database maintenance duties which are scheduled in certain periods. Shortening the query durations of the stored procedures used in the system is aimed.

AGBEP's database has three different login users and four different schemes. Using login users in database provides data security and consistency. First user is *admin* user. This is the most powerful user in the database. Admin user handles all transactions in all tables such as update, delete, and create. Briefly, admin has all permissions in database. The second one is *salesman* user. Salesman has permissions only in customer tables and order tables. Salesman only has edit operation. Last login user is *customer*; which can access customer tables and order tables.

Four different schemes (admin, customer, product and eMallSetup) are created to divide the database into parts and increase security. Each scheme in e-Mall database contains the related tables. The defined users in the database are authorized according to their jobs and their access is associated with the related scheme and it is kept under control. If we give an example, a user defined for a member shop manager who cannot access the eMallSetup scheme. So, a configuration change by mistake is prevented.

4. AN AGBEP APPLICATION: A DIGITAL WAREHOUSE

A sample application of AGBEP is staged as a Digital Warehouse which specialized on photography and its accessories are displayed in Fig. 15. The web project where the AGBEP platform is integrated is designed as to give all services that an e-Mall offers. Standard e-commerce processes such as showing product lists, giving detailed product information and search and buying modules are accomplished.

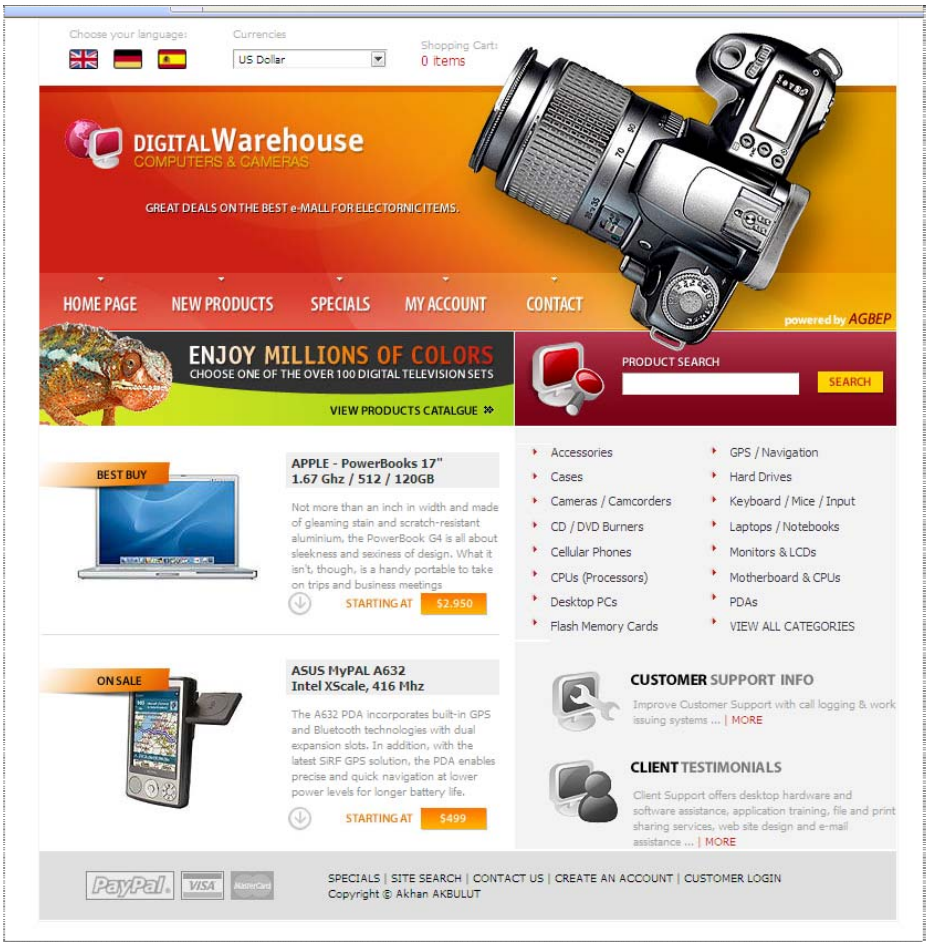


Fig. 15. Digital Warehouse web site as The Implementation Subject.

To access AGBEP's working parts, the member shop's manager gives his/her user name and password. We reach the login screen by choosing MY ACCOUNT on the main menu.

If we logged on before or if there is an active session, it is possible to reach feed uploading screen without facing the login screen in Figure 16. We access the system by giving the member shop manager's user name and password -which are defined in the system- on login screen.

After the successful entrance to the system, the member shop copies the xml feed which contains the products' information that they want to add to the system, to the file server by using the interface and starts agents' working chain.

An example of an XML feed is shown in Fig. 16. The tag which is at the top level of feed contains the member shop's name information. Each product under the member shop differs from each other with -Product- tag. The Category, Name, Description, Price and URL information is stored by the same name tags of the products which we want to add to the system under product tags.

The information in the Category tag is not taken into consideration. This product's best displayed category is determined with the analysis of keywords in the Description tag.

As shown in Fig. 17, all the products in the feed are matched with the best fit category to e-Mall by AGBEP. After this step, the member shop's products will be exhibited under the determined category in e-Mall and will be ready to purchase.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <akhan>
- <PRODUCT>
    <category>Photography</category>
    <name>Canon EOS 450D 18-55 IS Lens Kit</name>
    <description>12.2 Megapixel 7200 RPM Seagate 200 Gigabyte</description>
    <price>669.99$</price>
    <URL>http://www.digital-cameras.com/xpp-canon_eos_450d_18_55_is_lens_kit.html</URL>
  </PRODUCT>
- <PRODUCT>
    <category>LCD TVs</category>
    <name>SAMSUNG LE-37M87BD 37 94cm LCD TV.</name>
    <description>Component, 2 Adet Scart, 3 Adet HDMI, PC Giris, S-Video, Composite Video, RS232, DVB-T, Kulaklik cikis
      Var L/R, Optik ses, HDMI Version : Version 1.3 DVB Tuner : Var, LCD TFT aktif matris</description>
    <price>2536 USD</price>
    <URL>http://www.gold.com.tr/urunozel.asp?ID=39880</URL>
  </PRODUCT>
- <PRODUCT>
    <category>SAbit Diskler</category>
    <name>WD 320 GB 7200RPM 8MB SATA2</name>
    <description>320 GB 7200 RPM Veri Transfer Hizi 300 MBps</description>
    <price>88 USD</price>
    <URL>http://www.gold.com.tr/urunozel.asp?ID=40581</URL>
  </PRODUCT>
</akhan>
```

Fig. 16. Example XML Feed.

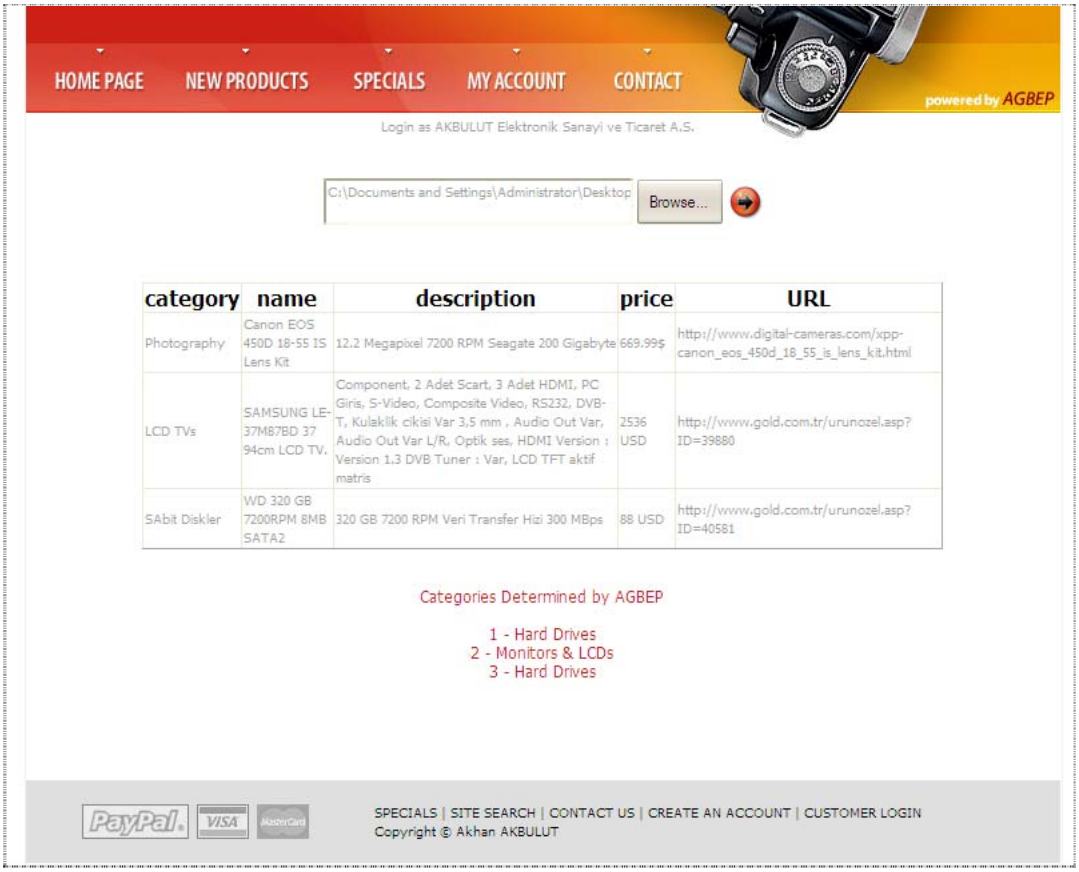


Fig. 17. Category Identification Screen.

5. CONCLUSIONS

5.1. Overview

In this study, we developed an agent based e-commerce platform, AGBEP using intelligent agents, hashing techniques and clustering methods. For the categorization of the product information which we want to add to the platform, system is carried out and is shown over e-Mall in an autonomous way. As the product information is mentioned in detail by member shops in Section 3, it is possible that it can be sent in the requested format and its integration to the system is maintained by the agents that function over AGBEP's agent server. In this process, after the keyword information of the products is determined, they are rendered suitable for clustering by hashing

methods. After the second level clustering process completed by accepting the correctness of the categories of the products inside the system, the categories of the products inside the feed are determined and e-Mall became ready for sales.

System performance has increased and data consistency has been achieved for AGBEP. During the development stage, we especially focused on the clustering process since it is the critical phase of the entire process. Also, the hashing process is another important phase for recovering product data before entering the clustering operation. By tuning these processes with intelligent agents, we created a full performance autonomous job for e-commerce content management necessities.

5.2. Future Work

In order to advance AGBEP, we need to concentrate on some other subjects. Currently, products are matched to the categories and sub-categories defined in the system's existing infrastructure. Moving forward, a decision-making mechanism could be added that implements a dynamic category matching process for a category not yet defined in the system. In the meantime, we advise that for a product that is far from the system's categories a new category should be added to the system.

To gain a higher performance, it is advised that the load balancing process in the web servers must be transferred to the physical load balancing devices. This solution is more expensive in terms of price, but works more efficiently than the Windows Server 2008's software load balancing service.

As a second step, XQuery [21] language can be applied so that the Inspector Agent can analyze XML product feeds with much higher performance. Extracting data from large XML documents causes difficulties for the entire system.

Finally, applying neural networks to our system for decision support mechanisms could be much more effective in getting rid of lack of intelligence in decentralized software systems, communication of software modules and system learning problems [22, 23].

6. REFERENCES

[1] Akbulut A., Yilmaz G., An Architectural Model for Content Management in E-Commerce Applications, *Proceedings of IADIS e-Commerce 2007 Conference*, 7-9 Sept., Algarve - Portugal, Page(s): 260-264

[2] Gelmez B., Sancar M., Yilmaz G., Akbulut A., An Autonomous Decision Support System Prototype For Suply Management In e-Commerce Systems, *Proceedings of IADIS e-Commerce 2008 Conference*, 25-27 July, Amsterdam – Netherlands, Page(s): 326-328

[3] Tang, O., Xie, F., A multi-agent system for e-commerce automation, IEEE CCECE/CCGEI, *Electrical and Computer Engineering, Canadian Conference 2006*, Ottawa Page(s): 514-517

[4] Shopping and Price comparison European Leader, http://www.kelkoo.com/

[5] Shopzilla - Comparison shopping online, http://www.shopzilla.com/

[6] Google Product Search, http://www.froogle.com/

[7] PriceRunner UK - Compare Prices and Find Best Deals Online, http://www.pricerunner.co.uk/

[8] Vakali, A., Catania, B., Maddalena, A., XML data stores: emerging practices, *IEEE Internet Computing*, Volume: 9, Issue: 2, Page(s): 62 - 69 2005

[9] Chan, C. Sun, A. Lim, E. Automated Online News Classification with Personalization, *The 4th International Conference of Asian Digital Libraries 2001*

[10] Yanagida, T., Miura, T., Shioya, I., Classifying News Corpus by self-organizing maps, Communications, *Computers and signal Processing IEEE Pacific Rim Conference 2003*, 800- 803 vol.2

[11] Xiao-Lin W., Cloete, I., Learning to classify email: a survey, *Machine Learning and Cybernetics, 2005*. Page(s): 5716 - 5719 Vol. 9

[12] Anagnostopoulos, I. Anagnostopoulos, C. Loumos, V. Kayafas, E. Classifying Web pages employing a probabilistic neural network, *IEE Proceedings 2004- Software*, Volume: 151, On pages: 139- 150

[13] Taeho J., Gwyduk Y., List based matching algorithm for classifying news articles in NewsPage.com, *IEEE International Conference System of Systems Engineering, 2008. SoSE '08.* Page(s): 1 - 5

[14] Bigus, J. Bigus Jn., Constructing *Intelligent Agents Using Java: Professional Developer's Guide Series*, John Wiley & Sons; 2001

[15] Lagus, K., Kaski, S., Keyword selection method for characterizing text document maps, *9th International Conference on Artificial Neural Networks-ICANN'99*, Edinburg, UK, pages: 371-376, vol.1 (Conf. Publ. No. 470)

[16] Gutmann, P., Naccache, D., Palmer, C.C., When hashes collide [applied cryptography], *IEEE Security & Privacy Magazine*, Vol. 3, pages: 68- 71, 2005

[17] Ordonez, C., Integrating K-means clustering with a relational DBMS using SQL, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, Issue: 2, on pages: 188 – 201, 2006

[18] Wang Z., Comparison of Four Kinds of Fuzzy C-Means Clustering Methods, *Third International Symposium on Information Processing (ISIP), 2010,* Page(s): 563 - 566

[19] eBay - New & used electronics, cars, apparel, collectibles, sporting goods & more at low prices, http://www.ebay.com/

[20] Microsoft Developer Network, http://msdn. microsoft.com/

[21] Funderburk, J.E., XML programming with SQL/XML and XQuery, *IBM Systems Journal*, Volume 41, Issue 4. Pages: 642 – 665, 2002

[22] Huhtinen, J., Utilization of neural network and agent technology combination for distributed intelligent applications and services, *Auditorium TS101*, Linnanmaa, on Nov. 4th 2005

[23] Aspect-Oriented Software Development Community & Conference, http://aosd.net/

VITAE

Akhan AKBULUT

BS in Computer Engineering *(İstanbul Kültür University, 2001)*, M.Sc in Computer Engineering *(İstanbul Kültür University, 2008)*, Ph.D candidate in Computer Engineering *(İstanbul University, from 2009)*. He has worked as senior/lead software developer and project manager in software sector and participated in many middle and/or enterprise size projects. He works in the Computer Engineering Department of Istanbul Kültür University. His current research areas are software engineering, web and mobile based application development and wireless sensor networks.

Asst.Prof.Dr.Lt.Col. Güray YILMAZ

He is an Assistant Professor in Turkish Air Force Academy. He received his BS, MSc and PhD degrees from the Istanbul Technical University Computer Engineering Department in 1991, 1995 and 2002 respectively. He teaches Operating Systems, Parallel and Distributed Systems and Special Topics in Distributed Systems. His current research areas are parallel and distributed systems, mobile agent systems, web services and wireless sensor networks.