An Agent-based Framework for Personalized Advertisement on Mobile Devices

Resul Şahin
Computer Science Department
Istanbul Technical University
Istanbul, Turkey
sahinres@itu.edu.tr

Nadia Erdogan
Computer Engineering Department
Istanbul Technical University
Istanbul, Turkey
nerdogan@itu.edu.tr

***Abstract*—Online advertising is the main profit channel of the internet companies. But, mobile advertising has increasing importance because of the widespread of mobile devices. Effective mobile advertising is the key factor for maximizing revenue when considering advertisers and subscribers benefits. So, An effective ads system take into account that mobile broadband, user time, personal preferences and information. This paper presents a framework to facilitate a flexible and intelligent personal mobile advertisement on mobile devices. The developed framework consists of a combination of expert systems and multi-agent systems. The personal agent communicates with the server agent of the multi-agent system, in order to discover appropriate advertisements to advise the user.**

Keywords-multi-agent sytems; expert systems; Jade

I. INTRODUCTION

Ads should be presented according to preferences of the user to improve the success rate in mobile advertising. This process consists of two main parts. First, user preferences are expressed using advertising domain information such as campaign start/end date, discount rate, product/service properties and so on. Second, the appropriate ones to user preferences must be determined from all the ads. However, this process is of a highly dynamic structure. Because, each user is interested in different ads. In addition, a user have to read to evaluate separately each ads because of text-based nature of ads. Therefore, this process is far from being goal-oriented in terms of advertisers, as well as the time-consuming activity for users. In this study, an agent based framework has been developed in order to identify the most suitable ads which can be interested by the user. Thus, the proposed framework is not only goal-oriented for advertisers but also provides time-saving for users.

A person can be interested in many domains permanently or temporarily. In addition to this, number of domain providers varies greatly from small to large. So, If the person is interested in a domain, he/she needs to register/investigate all providers sites. This task is so time consuming and also requires sharing personal information with each providers if the person wants to be up to date with future ads. Moreover, the person is always exposed to information mass from providers. When the person receives more information, each information is gained less attention by the person.

Each ad is for a specific product or service, so it will not be able to make a standardization between the ads. Representing a conceptual model of the product/service which it belongs can be a method of standardization of the domain. Once the conceptual model is defined, the model can be used by all of the ads related with that domain. Thus, text-based ads by using the conceptual model can be done faster and more goal-oriented filtering. Most of the studies on the ad filtering used text-based information filtering techniques. But in this study, an ontology-based technique is used for ad filtering.

Semantic web [1] is a set of paradigms and new technologies about the web. Semantic web does not intend to replace current web, it tries to complete the current web. Semantic web is a new research area so many researchers working about it. Some of the research areas are that; proposing new standards, developing tools, data mining, knowledge managements and web services.

Ontology is one of the key paradigms in semantic web. An ontology [2] formally represents knowledge as a set of concepts within a domain, and the relationships between those concepts. It can be used to reason about the entities within that domain and may be used to describe the domain.

Each ad belongs to the domain will be expressed in terms of ontologies in the developed system. The method of determining the ads which meet the user's preferences is also an other important point. An ad can be discovered with writing code whether or not it is appropriate the user's preferences. But, writing a piece of code for each ad have negative effects on system flexibility, usability and extensibility. Therefore, an expert system designed to determine the appropriate ads considering the negative effects on system. Because, expert systems are widely used in industry because of their cost-effective and fast responsiveness nature. An expert system can apply a domain if the domain knowledge can be modeled by the means of rules. Modeling domain knowledge requires co-working with domain experts and software engineers/analysts. When domain modeling is completed, it is easy update, add, delete defined rules on the fly.

This paper presents an agent-based framework for personalized advertisement on mobile devices. The system architecture is shown in Fig.1.

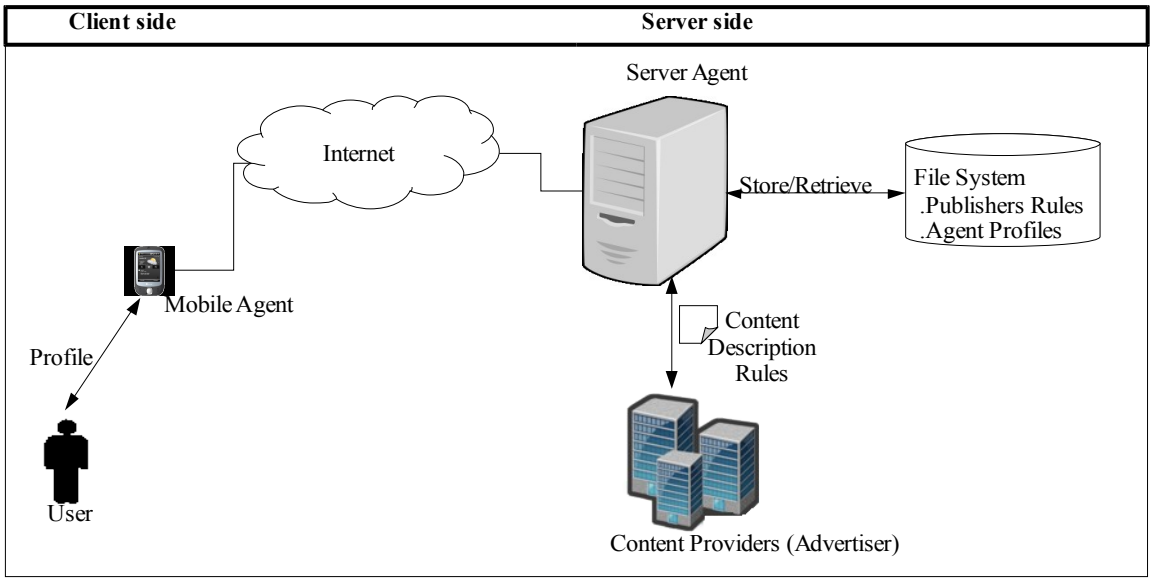

Figure 1. Final system architecture.

The rest of this paper continues with Section 2, which briefly summarizes recent works. In Section 3, the motivation of this thesis is described. Next, Section 4 explains design and implementation details of the proposed system. In Section 5, the system is tested with under sample scenario considering to applicable and performance. Finally, the paper finishes with a conclusion.

II. Background and Related Work

There are several previous research studies about rule-based agent systems [3]. One of them is that, developing a flexible and extensible workflow automation tool kit for the healthcare domain by combining multi-agent and rule-based technologies [4]. All contributors of healthcare domain is modeled with JADE agents in Vogt's research and rules defined with Jess rule engine [5].

If the data is unstructured form, information filtering techniques is more appropriate for filtering. Ontology-enhanced information filtering agent [6] applied information filtering techniques using WORDNET's [7] ontology. But, when the information is uniform, problems turns out the content-based filtering problem.

To realize an appropriate content-based subscription system, design considers application major requirements such that efficiency and expressiveness which are generally opposite concepts. Topic-based is picked up when the efficiency is more important than expressive. But, enabling to more complex subscriptions often requires expressiveness where this type is called content-based publish-subscribe. It is clear that, this type of systems requires more performance than other type. So, solving such a problem, Rete is used that, Rete [8] is a more powerful and well-known algorithm for pattern matching. Rule-based system such as Jess, Clips [9] and Drools [10] use Rete implementation for pattern matching. A scalable publish/subscribe system with expressive and flexible semantics [11] is developed with rule-based approach using Jess.

There is an another research about personalized ads where mobile advertising platform based on three aspects: context, content and user preference. The system makes utilization of velocity-detection and content-match to allocate personalized ads [12]. In Chang&Huo's work, content-match leverages on IF techniques.

III. Motivation

There are also researches about personalized ads using IF filtering techniques. But, In this work, I want to apply a rule-based approach to same topic.

The researches show that, rule-based systems can be applicable to publish/subscribe systems. On the other hand, an advertisement can be modeled by rules where the rules are used for content-matching by an expert system. So, an agent which gather user information about advertisements by the means of ontologies. And this collected information is used by an expert system to discover the most appropriate ads for the user. Of course, personal decision process is very complicated and continuous. But, an intelligent agent can save time when person consumes at information discovery time and reduce information traffic on broadband.

To show that, agent based model augmented with ontology is designed and developed. This paper presents an agent-based framework for personalized advertisement on mobile devices.

IV. Personal Agent

The system is implemented on JADE(Java Agent Development Framework) [13] which is fully compliant with FIPA [14] specifications and has support for platforms with limited resources, such as mobile devices. To perform pattern matching, a rule-based system Drools is chosen, which is open source Java implementation of Rete algorithm.

Mobile agent is implemented on Android using mobile version of JADE which is called JADE-Leap. All ad content/ knowledge is represented with ontologies. Client or subscriber fills its interests using ontology based forms where the forms are dynamically generated from the ontology.

A. System Participants

Two type of agents implemented in this work called client agent and server agent. A client agent is owned by specified user and runs on mobile device which has Android operating system. In general, client agent is an Android application developed with JADE-Leap library. The packages on agents is depicted in Fig.2.

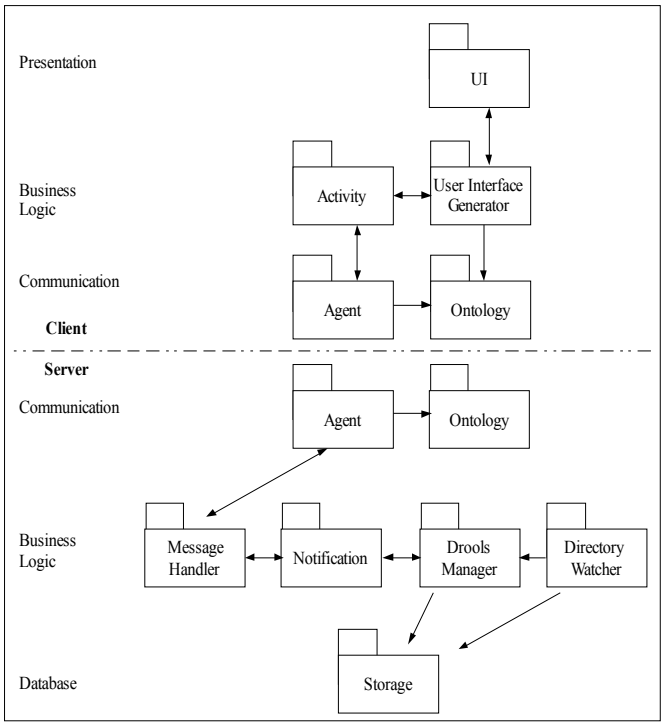

Figure 2. Packages on agents.

The following sections give detailed information for each type of agent.

1) Client Agent

a) Agent package: This package contains the procedures for the establishment of the agent based infrastructure on Android OS. Jade works as a service in Android application. When the application is opened, the client registers main-container by entering the username. In addition, given username can be used by a single user in the system.

All the communications with the server agent carried out with appropriate ontology using ACLMessage(s). Client agent creates the behaviours which will be used for communication with the server agent. Major behaviours are that;

- CyclicBehaviour handles incoming messages from the server.
- OneShotBehaviour makes instant request to take appropriate ads to client preferences.

b) Ontology package: If there are many ad providers and clients in the system, a common denominator is required to achieve standardization. This denominator can be provided with ontologies which became more important with the Semantic Web. Ontologies can pre-defined for each domain and can be shared with service providers and clients. Jade ontologies can be defined on Protege with an add-on [15].

Ontology development starts from by defining its domain and scope. In this study, flyOntology has been developed with required specifications considering flight domain and ad scope.

As a result, ontologies are used for conceptual abstraction of a domain. This package is common between client agents and server agent.

c) User interface generator package: This module is very important in terms of the applicability of the designed system. Providing for content-based filtering through each user agent among ads, data entry screens must be present.

Developing code for each domain is not eligible as to cost and time. So, ontology is used in a standard way for build a dynamic screen.

This package generates the screen from the domain-related ontology class using Java reflection. Information entered by the user is got via reflection from dynamically generated screen. The retrieved data are that each field value and the operator which is entered for this field. This information is written into ontology class wrapper object and sent to the server. Ex: Assume that campaign start date is a field in the ontology. And, the value equals "08/05/2012" and the operator is chosen as ">" which means after the given date.

d) UI package: As mentioned above, screen is rendered dynamically from ontology class. This package contains each sub-component classes that make up the form. A form consists of different number and sequence of sub-component classes on the ontology. Some sub-components are that; UiEditBox, UiDatePicker, UiCheckBox and UiSelectOne.

For example, UiEditBox is a sub-component of the screen which generates label, value and operator fields on the screen. In addition, it handles inputs entered by the user.

e) Activity package: This package contains of Android activity classes such as listing defined domains and showing received notifications to user.

2) Server Agent

This agent runs on the server and serves notification service to clients in accordance with the client's preferences. A notification is sent to the client when there is a matching between a product/service and client's preferences in all ads. Sending notification may be made instant or predefined time periods.

The content of the each package is described separately below.

a) Agent package: The server side is implemented as a standard JADE agent to communicate with clients to respond to their requests. When it comes any request from a client, the

server checks whether the appropriate ads using Drools engine.

b) Ontology package: This package is common between client and server agents.

c) Drools manager package: This package creates and manages Drools Session module. Drools session performs rule-based matching between client's preferences and ads. This module creates knowledge-base of the domain by reading all ad providers information. All provider information is located in the drl files on the disk. The module adds a fact into Drools session corresponding to each notify request. The fact represents client's preferences related with current domain. Drools session is executed and notification results is returned to client. This process is logged to disk through custom working memory and agenda listeners for later inspection.

d) Directory watcher package: All providers (advertisers) information is located on the server's file system. Each provider has a separate directory for stored data. When the system is running, new providers can be added or some of them can be left. In addition, the advertising providers change rules when the system runs. These changes needs to automatically detected and knowledge-base should be updated without restarting the system. So, the file system is checked at regular intervals with a dedicated thread. If a change is found, knowledge-base is automatically updated. Thus, there is no need to restart the system when any change occurs.

In summary, this module is developed as a separate thread to provide the knowledge-base up-to-date.

e) Message handler package: Each incoming ACL message is transformed to application object and the appropriate handler is invoked by the this package. Required message is returned to the clients when a protocol error occurs.

f) Notification package: This package contains classes for notification service. A domain object which is composed with client's preferences is added to the Drools session. The session is executed, and activated rules are recorded. Result set is returned to the client via new ACL message. In addition, this package logs the steps of the pattern-matching to the disk. The response to each request can be monitored later through recorded logs.

V. Sample Scenario

This section shows how the developed system works on a scenario. The target is that, notify the clients when a campaign is found according to their preferences. Clients can change their preferences over time.

When developing an ontology, the requirements of application must be considered. An ontology is developed for flight domain in sufficient depth and features. The developed fly concept is given in Fig.3.

```
public class Fly implements Concept{

   @UiAttributes(name="Company Name", order = 0)
   private String company;
   @UiAttributes(name="Flying Type", order = 1)
   private FlyType flyingType;
   @UiAttributes(name="Departure", order = 2)
   private String departurePlace;
   @UiAttributes(name="Arrival", order = 3)
   private String arrivalPlace;
   @UiAttributes(name="Campaign Start", order = 4)
   private Date startDate;
   @UiAttributes(name="Campaign End", order = 5)
   private Date endDate;
   @UiAttributes(name="Cost", order = 6)
   private float cost;
   ....
}
```

Figure 3. Fly concept in the ontology.

Fly class is formed from flight information and hierarchically derived from Concept in the ontology. UiAttributes annotation is used in the class definition will be used during the creation of the user interface. And, the annotation specifies the name and the order of a sub-component on the screen.

The campaign conditions of the three companies are given below;

- Anadolu Jet: Flights from Istanbul to Van, Samsun, Hatay announced $70 between the dates of June 5-25, 2012.
- Izair: Istanbul-Izmir flights announced $55 between the dates of June 5, 2012 and July 5, 2012.
- Pegasus: All domestic flights announced $60 between the dates of June 1, 2012 and July 1, 2012.

To assess the text-based ads in developed system, the information in ads is expressed as Drools rules. Rules can be written by analyst or programmer. Corresponding Drools rules are given in Fig.4 below for each campaign.

```
anadoluJet.drl
 rule rule_AnadoluJet
  when
   $fly : Fly(departurePlace == "Istanbul",
   (arrivalPlace == "Van" || arrivalPlace == "Samsun" || arrivalPlace == "Hatay"),
    eval(startDate  != null && startDate.after(new Date(2012,6,5))),
    eval(endDate != null && endDate.before(new Date(2012,6,25))), cost <= 70)
  then

izair.drl
 rule rule_izair
  when
   $fly : Fly((departurePlace == "Istanbul" && arrivalPlace == "Izmir") ||
  (departurePlace == "Izmir" && arrivalPlace == "Istanbul"),
   eval(startDate  != null && startDate.after(new Date(2012,6,5))),
   eval(endDate != null && endDate.before(new Date(2012,7,5))), cost <= 55)
  then

pegasus.drl
 rule rule_pegasus
  when
   $fly : Fly( flyingType == FlyType.DOMESTIC,
   eval(startDate  != null && startDate.after(new Date(2012,6,1))),
   eval(endDate != null && endDate.before(new Date(2012,7,1))), cost <= 60)
  then
```

Figure 4. Drools rules for sample ads.

The client installs agent-based android application to discover ads on its mobile device. The client interface is given in Fig.5 for flight domain. The client enters the flight preferences using the screen.

Figure 5. Client user interface for flight domain.

The client can specify the restrictions on its flights. In the above example, the client deals with flights from Istanbul to Van between dates of June 10-14, 2012. The price limit of the client is $65. If there is a campaign which meets the conditions currently or in the future, a notification message will be sent to the client about the campaigns. Currently, there are two campaigns which satisfy the client conditions on the system. They are AnadoluJet and Pegasus campaigns. The client receives two campaigns notification as shown in Fig.6.

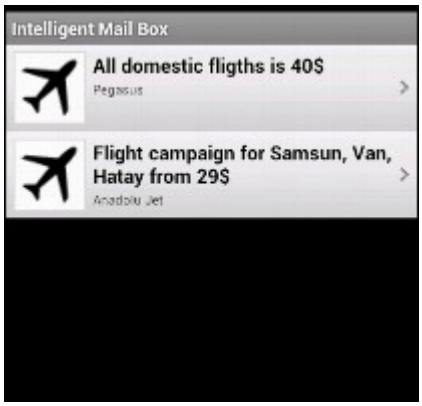

Figure 6. Intelligent mailbox of the client.

A. *Performance*

Performance of proposed system is tested. The average response time of the system was measured with different workloads. All measurements was done on personal computer which has these features; 2.27 GHz dual core Intel Core i5 CPU, 4 GB RAM, 64-bit Windows 7 OS.

All clients made requests simultaneously to the system. Each request is sent by the test agent within its JADE container. The average response time increased when the number of simultaneous request is increased. If the acceptable response time is 50ms for the system, the test environment will serve maximum 400 simultaneous requests. When simultaneous clients was increased, the system slowed down because of the context-switching cost of the large number of threads. To overcome this condition, the system can be improved with creating a thread-pool to limit the number of maximum simultaneous request at a time as application servers do. Test results is showed in Fig.7.

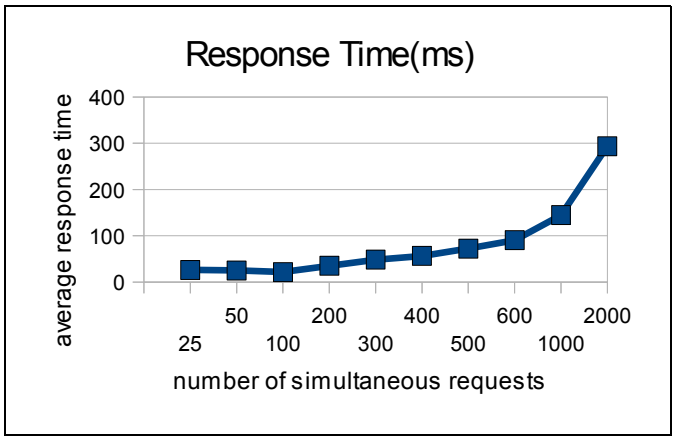

Figure 7. Results of the load tests.

It is obvious that, the performance of the system is dependent not only on the Drools engine itself, but also on how the rules are written and how the domain model looks like.

VI. Conclusion

In this study, an Agent-based Framework for Personalized Advertisement on Mobile Devices is developed. There are two tasks for adding new domains to the system. First task is defining the new ontology for the target domain. The second task is to write Drools rules, and then adding them to knowledge-base of the system. The system offers a flexible, extensible framework using the expert systems and the agent-based technologies. The system is implemented on JADE. So, it is fully FIPA compliant.

Using a rule-based approach in the developed system, the mobile device owners will have the customizable service which discovers tailored ads for their preferences. That allows clients to quickly access to the ads which are more accurately for themselves. The system also prevents the users from being lost among ads. In addition, saves users time and bandwidth in the ad discovery process.

In future work, the current system will be improved using client instant data such as location and velocity. So, more intelligent agents can be achieved in the future. As a result, a campaign discovery tool is developed which relies on rule-based system. The tool is totally guided by the user, thus this

approach differentiates the proposed system from the existing campaign suggestion systems.

References

[1] T. B. Lee, J. Hendler, and O. Lassila, "The semantic web," Scientific American, vol. 284, no. 5, 2001, pp. 35–43.

[2] Wikipedia contributors. (2012, Sep 8). *Ontology (information science)* [Online]. Available: http://en.wikipedia.org/wiki/Ontology_(information_science)

[3] A. Beneventi, A. Poggi, M. Tomaiuolo, and P. Turci, "Integrating rule and agent-based programming to realize complex systems," WSEAS Transactions on Information Science and Applications, vol. 1, no. 1, 2004, pp. 422-427.

[4] J. Vogt, "A rule-based solution supporting intelligent and adaptive agents," M.S. thesis, Software Engineering Group, Department of Informatics, University of Fribourg, Switzerland, aug. 2008.

[5] E. F. Hill. (n.d.). *the Rule Engine for the Java Platform* [Online]. Available: http://www.jessrules.com/jess/index.shtml

[6] K. M. Sim, "Toward an ontology-enhanced information filtering agent," SIGMOD Record, vol. 33, no. 1, 2004, pp. 95-100.

[7] G. A. Miller, "Wordnet: an on-line lexical database," Int. J. Lexicography, vol. 3, no. 4, 1990, pp. 235-312.

[8] C. L. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," Artificial Intelligence, vol. 19, no. 1, 1982, pp. 17-37.

[9] Wikipedia contributors. (2012, Aug 19). CLIPS [Online]. Available: http://en.wikipedia.org/w/index.php?title=CLIPS

[10] Drools. (2012). The Business Logic integration Platform [Online]. Available: http://www.jboss.org/drools

[11] D. Matheson, "A rule-based approach to publish/subscribe," M.S. thesis, Department of Electrical and Computer Engineering, University of Toronto, 2004.

[12] C. H. Chang, K. H. Huo, "Mobile advertising: triple-win for consumers, advertisers and telecom carriers," ACM SIGIR 2011 Workshop on Internet Advertising, National Central University, in press.

[13] JADE Home Page. (2012). [Online]. Available: http://jade.tilab.com

[14] FIPA. (2012). The Foundation for Intelligent Physical Agents [Online]. Available: http:/www. fipa.org

[15] C. J. van Aart. (2008). Ontologybeangenerator [Online]. Available: http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator