ACTIVE ADVERTISEMENT IN SUPERMARKETS USING PERSONAL AGENTS

Jarogniew Rykowski
Department of Information Technology, Poznan University of Economics
Mansfelda 4, 60-854 Poznan, Poland
e-mail:rykowski@kti.ae.poznan.pl

Abstract: In this paper we present our approach to active marketing in supermarkets, with an extensive use of radio-frequency identifiers (RFIDs) of goods and locations, and personal, mobile communication devices of clients. Our solution is based on personalized filtering of RFID-related marketing info by the use of software agents. As an implementation base we use the Agent Computing Environment (ACE) framework. ACE agents are partially executed in clients' mobile devices, and partially in the local area network of a supermarket. These agents are able to filter broadcasted marketing information according to detailed user needs, technical restrictions for end-user device, date and time, previous contacts, etc. Several strategies are proposed to use ACE agents to support everyday shopping in a supermarket, including maintaining shopping lists, personal marketing firewalls, and monitoring for changes.

Key words: Software agents, personalization, targeted marketing, RFID

1. INTRODUCTION

Recently we observe two basic trends in modern economy: the globalization, and mass applying of new information and telecommunication technologies. The globalization stands for world-wide market competition regardless time and geographical limitations. Fighting off this competition forced remarkable business changes. Rather than cutting a price, that is already lowered to the maximum extent, one has to compete by other means, for example by broadening and personalizing an offer. However, mass production and selling of personalized goods and services is putting out a great technical and organizational challenge. To this goal, modern technologies must be applied, at each stage of the production and distribution process, including design, logistics, marketing, etc. All these technologies require an efficient information flow and flexible system architecture. Recently, mass tele- and radio-communication systems seem to be the most promising in that domain, mainly mobile telecommunication systems from a client point of view, and bar codes and radio-based automatic identification of goods (RFID identification [16]) for supplying mass production and logistics. Many world-wide companies targeted on mass clients, as owners of supermarket networks for example, widely applied such technologies to cut overall costs, to improve contacts with clients, etc.

So far, a process of applying the above-mentioned modern technologies was transparent for ordinary users – market clients. These technologies were reserved for internal production, logistics, and distribution stages, with limited interaction with end-users [12]. However, there is huge clients' expectancy to change this situation as a result of mass introduction of personal communication devices and portable computers. Returning to supermarkets, the clients are interested in using mobile information technologies to support their everyday shopping. However, so far mass usage of personal clients' devices is not supported by supermarkets, except for very few examples and testbeds, e.g., automatic payment systems in WalMart network [7], and "intelligent" shopping carts [15].

In this paper, we propose to use two modern technologies: personal software agents, and radio-frequency identifiers (RFIDs), to improve the way of shopping in supermarkets. We propose to support the whole process of choosing goods (marketing and comparison of different offers), finding them in the supermarket, and finally buying them. At the same time, our technology may be used for distributing marketing information among clients, as well as for improving internal management of a supermarket. The main goal of this paper is to propose a framework for targeted marketing inside a supermarket, based on RFIDs of goods and locations, and intelligent filtering of incoming marketing information by client's personal (mobile) devices. We propose to use agent-based ACE framework, developed by us and already used with success in many application areas, to create an efficient and safe environment for preparing, storing and executing agents.

The remainder of the paper is organized as follows. In Section 2, some basic information about radio communication and RFID technology is given. In Section 3, agent technology is described, and the ACE (Agent Computing Environment) is presented. In Section 4, an overall architecture of the system is presented. In Section 5, some basic strategies are given of using ACE agents for targeted marketing and personal filtering of marketing information. In addition, some tips are given of using ACE agents for improving supermarket management. In Section 6, a short comparison with some related work is presented. Section 7 concludes the paper.

2. RADIO COMMUNICATION AND RFID IDENTIFIERS

Radio communication systems were traditionally divided into two basic groups: systems devoted to human-to-human voice communication, and computer networks. The human-related communication networks evolved during last twenty years from small analog networks with proprietary solutions to world-wide GSM standard. At the same time computer networks evolve, by introducing new protocols and hardware for radio-based data transfer, mainly for personal (BlueTooth) and local networks (WiFi). Recently, we observe a trend of mixing the above mentioned communication systems into a single, universal, multimedia network (the UMTS idea). The same end-user device and networking equipment is used for voice (recently multimedia) communication, and data transfer. Thus, a user is able to communicate via different standards, according to current situation, environment, technical and economical limitations, etc.

Apart radio-communication systems for humans, there is a large application area of using such systems for automatic wireless connections among different devices. Among others, radio frequency identification (RFID) is one of the hottest research topics. RFID is a generic term that is used to describe a system that transmits the identity of an object or person wirelessly, using tags accessible by radio transmission [16]. A typical RFID tag consists of a microchip attached to a radio antenna mounted on a substrate.

So far, RFID tags are not compatible with Bluetooth/WiFi communication standards. Thus, some additional equipment (a specialized reader) is needed. Such a reader must be mobile, thus it is usually equipped with a radio communication link to a personal/local computer network (WiFi/DECT). The RFID readers are not cheap, however, despite RFID tags, such readers may be reused. For example, a Bluetooth-compliant two-system reader may be applied to a shopping cart in a supermarket. Then, an owner of a mobile phone with Bluetooth connection may take full profits of RFID identification as long as he/she is close to the cart. Note that the connection and communication may be fully automated, and the data transfer is free of charge. Note also that the network is not overloaded due to small size of Bluetooth cells and thus limited number of local signal (frequency) conflicts.

Shopping carts with Bluetooth(WiFi)/RFID readers may be also used by supermarket staff, to transparently track clients and their shopping customs, timings, to compute some anonymous statistics, to improve internal management, etc. Moreover, such carts may automatically inform clients about current price of all the collected goods, send some advertisement info while passing by some goods/locations, support final checkout, etc.

3. AGENT TECHNOLOGY

In our approach, we used a classical definition of a software agent, as presented in [4, 6, 17]. A software agent is a program, executed at a given place, characterized by: (1) autonomy – agents process their work independently without the need for human management, (2) communication – agents are able to communicate with one another, as well as with humans, and (3) learning – agents are able to learn as they react with their environment and other agents or humans. As follows from the above definition, an agent may be programmed by its owner, thus allowing unrestricted personalization of behavior of this agent [1]. Agents may be executed in different places, according to owners' needs and possibilities of the end-user hardware [5].

The ACE framework is based on a set of distributed Agent Servers [8-11], each of them capable of storing and executing software agents. The agents may be moved among Agent Servers. There are "light" Agent Servers with limited functionality to be executed in a "thin" hardware/software environment (e.g., mobile phones), and "thick", massively used Agent Servers located in stationary network hosts. The "light" servers are mainly used for executing individual agents of an owner of a mobile device, while the "thick" ones are used by many users in parallel, usually to access certain services, external software systems, and public communication channels.

There are two basic classes of ACE agents: public System Agents, and Private Agents. Public *System Agents* SAs are created by trusted users (usually system designers), to be used in a mass manner by many users, providing information in a standardized form and with optimum effort. As overall efficiency is of primary concern, SAs are programmed in Java. A way of usage of a given SA cannot be changed by an ordinary user, however, it may be parameterized during the invocation [8, 10].

The *Private Agents* PAs are created and controlled by their human owners. Unless directly ordered by its owner, the agent cannot be accessed by any other agent. For private agents, the main problem is to achieve a reasonable trade-off between overall system security and a need for remote (i.e., server-side) execution of user-defined, thus „untrusted" (from the local administrator point of view) code. Several restrictions and limitations must be applied to user-defined code, protecting the system from (intentional or accidental) damages. Thus, a specialized language is proposed to program agent behavior, based on XML and equipped with several non-standard mechanisms like run-time monitoring of CPU time and memory allocation [8]. The language is of imperative type, thus allowing much wider personalization of the agent code in comparison with the classical declarative approach. XML-programmed private agents may invoke huge library of on-site, residential, Java-based system agents: communicators, services, brokers to external software systems, tools and utilities, etc. Usually, a small private agent, being a "light" mobile entity, is able to use (i.e., execute) several system agents, to achieve different goals. From the user point of view, the system is effective and powerful, and even small private agents are "intelligent" enough to fulfill complex requirements. From the system point of view, private agents executed at server-side do not pose a threat to local environment and other agents.

A typical Agent Server is equipped with several specialized system agents, so called input/output gateways, able to communicate with an external world (including other Agent Servers, local and remote software, and humans) via communication channels of different type and purpose. In general, two basic types of human-agent communication gateways are available: textual and Web-based. A *textual gateway* is able to exchange flat (unformatted) text messages, usually among humans and agents. Physically, textual gateways may use such means as an e-mail connection, SMS/MMS connection with a telecommunication network, a voice gateway, etc. Once sent by a textual message, an ACE agent may act as a chatterbot, analyzing the message via keyword extraction and analysis [9]. The chatterbot interface is useful for non-advanced users, and for users temporary handicapped due to limited hardware possibilities and communication costs.

Web-based gateways are used to access an agent via a WWW/WAP page, and from specialized ACE applications. For semi-automatic formatting of both contents and presentation of the data to be sent, XSL-T technology was adopted with XSL transformations defined in a personal manner and stored in private agent variables [11]. To improve data formatting and presentation, automatic detection of end-user device may be applied, allowing auto-adjustment to the availabilities and technical possibilities of both communication means and end-user devices.

Gateways to external data sources are mainly used for automatic monitoring of information changes. As a change is reported by an external data source, a gateway invokes a selected agent. The agent may next pass the notification about "interesting" changes to user(s), via certain telecommunication gateways. What is "interesting" for the user is programmed by him/her in the code of the private agents [8]. Thus, a set of user's agents is an "intelligent", personalized filter of changes of monitored data.

4. SYSTEM ARCHITECTURE

Overall system architecture is presented in Fig. 1. The architecture consists in using a set of Agent Servers: mobile, light-weight Agent Servers in clients' mobile devices, and a public Agent Server of the supermarket.

The personal Agent Servers are used to execute private agents only. The public Agent Server is able to execute both private and public (system) agents, by the use of two distinguished agent pools: a user pool, linked with clients' personal devices and shopping carts, and a system pool, linked with external software systems of a supermarket (and possibly the Internet).

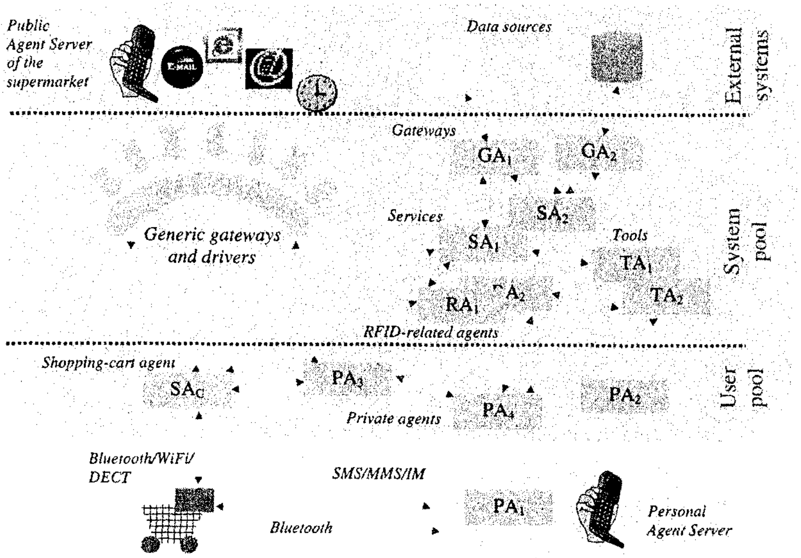

Figure 1. Layered system architecture

The *system pool* of the public Agent Server is devoted to four basic classes of agents: system gateways, services, tools and utilities, and agents related with RFID tags. *Gateways* are specialized agents able to connect with an external software system (cf. Section 3). *Public services, tools and utilities* are system agents prepared by the supermarket staff to be massively used by the clients' agents. These agents are characterized by a stable, well-defined functionality. One of the system agents may act as a directory-agent, providing implementation and access details about other agents. Some agents may be also related with supermarket services, information sources, databases, Internet access, etc.

The public Agent Server is linked with a centralized database of RFIDs related with descriptions of products/goods/locations. The database is usually located inside the local area network of the supermarket, however, it may be accessed via a public communication network as well. The basic functionality of the database is to provide a description/characteristics/ marketing info about a product/piece related with given RFID tag. This basic functionality may be extended, for example by searching for "similar" products/goods, finding an RFID identifier according to some keywords (textual characteristics) provided by end-users, etc. A WAP/WWW server may be provided to access the database from the Internet (e.g., using GPRS radio-connection and WAP/WWW interfaces of a mobile PDA/phone).

There is a possibility to access other Internet/extranet/intranet data sources, using agent-gateways of the public Agent Server as brokers and wrappers. Note, that the outcoming network traffic is controlled by a system agent, thus an access to public resources may be (automatically) moderated.

RFID agents are directly related with RFID tags, and indirectly, with goods and locations. A typical RFID agent is uniquely identified by its corresponding RFID "serial number". The agent is responsible for giving some information about the product/piece/location related with the RFID tag. To this goal, the agent is able to contact an internal database of RFID tags of the supermarket, and, if needed, the Internet.

The *user pool* is devoted to store and execute two basic classes of agents: agents related with shopping carts, and private agents of supermarket clients. Each *shopping-cart agent* is related with a single radio device located inside a given cart. The device is able to communicate with RFID tags (usually a proprietary supermarket standard), as well with mobile devices of supermarket clients (using Bluetooth and/or WiFi). The agent starts to work once the cart enters the supermarket, and is stopped after client's checkout. The main role of this agent is twofold. First, the agent is used as a broker between the RFID tags and the rest of the system. The notifications about detecting RFID tags passing-by are sent to shopping-cart agent, and in turn, to private agents of the client. It is up to the user agents to use such notifications or to ignore it. Second, the agent is a wrapper to the communication between the client's mobile device and the agents from the user or system pool of the public Agent Server. The client may register its mobile equipment in the shopping-cart communication device, e.g., by establishing a Bluetooth link. After registration, the client is able to register and execute his/her personal agents. Personal agents are identified by the use of a physical identifier of an end-user device – a network (usually MAC or Bluetooth-ID) address of a portable/pocket computer, and a phone number for mobile phones. Such way of identification ensures that (1) the identifiers of private agents are unique, (2) the identifiers are stable, and they survive a time of single shopping. Thus, the same identifier may be used next time the client is entering the supermarket. This creates a possibility of using resident, autonomous agents, monitoring selected goods/prices all the time and informing clients if anything "interesting" happens, even if the client is not visiting the supermarket at the moment (cf. Section 5).

Mobile *private agents* are able to move themselves, and they change their position as their owners move (both globally, from a supermarket to another one, and locally, inside a supermarket, from one Agent Server to another). However, a mobile equipment is not powerful enough to execute such complicated software, thus most of the private agents must operate in the stationary network, as close to their owners as possible, to limit the network traffic and system response time. For example, a private agent registers in the supermarket network once its owner enters the supermarket. During check-out, the agent either moves back to the private equipment, or freezes itself waiting for the next visit of the client. Once the client is back, the agent is automatically awaken and ready to serve client's requests.

In contrast to public agents related with system services and RFID tags, personal agents are not characterized by well-defined, stable functionality. Instead, a detailed functionality of a given private agent is under exclusive control of the agent's owner. Thus, we cannot provide here a detailed description of agent's architecture. Instead, we propose a strategy of using private agents for active marketing and on-line comparison of goods (cf. Section 5). However, this is up to particular clients to decide whether to use this strategy or not. The ACE framework is open for introducing any number of other strategies, individualized for both clients and products/producers.

For some less-advanced users with limited ability to program individual agent's behaviour, a standard agent may be created, e.g., implementing the above-mentioned strategy. More advanced clients, however, have a possibility of adjusting the strategy to their own expectations and current requirements. The experts are able to implement their own strategies, fully controlling agent's behaviour and state (cf. Section 3). Note that standard agents may be created automatically once the client is entering the supermarket, and disposed while he/she finishes shopping. Note also that such agents may use some supermarket's hardware only, as for example "intelligent", interactive shopping carts with LCD displays.

4.1 Using RFID tags

Unlike typical applications for logistics and management, we propose to use RFID tags to identify not only goods, but also selected locations inside a supermarket. The standard solution with simple RFID tags applied to goods only is quite expensive, and may generate huge network traffic. First, simple RFID tags cannot be reused, for obvious reasons. Second, for lots of pieces of the same product laying on a shelf, detailed serial numbers of these pieces are of no use, the client is rather interested in getting some general info about the product. However, when lots of tags of the same type are trying to register to a single reader providing similar information, continuous, huge network traffic is generated. Thus, an additional filtering mechanism is needed, and some generalization (i.e., a mapping from identifiers of pieces to identifiers of products) must be provided. Taking into account the above limitations, we propose to use simple RFID tags only for single-piece, usually expensive products, as for example TV sets, audio-video equipment, etc. To identify "ordinary" goods we propose to use RFID tags related with locations, i.e., places inside a supermarket where these goods are displayed on the shelves. In such case, a single tag is used to identify any piece of a product, and any piece of the same product is treated in the same manner.

There are several advantages of linking RFID tags with locations rather than pieces of goods. First, the tags are reused, thus the whole system is quite cheap. Second, the reused tags may be technically more complicated than simple RFID tags. In particular, a proprietary radio communication may be replaced by standardized networking connection (Bluetooth, WiFi, etc.). Thus, such a tag may be accessed by standard mobile equipment, including most of PDAs and mobile phones, without a need for dedicated RFID reader. In the case of simple RFID tags related with goods, an additional hardware broker is needed, usually located in a shopping cart. Such broker is responsible for contacting RFID tags, eliminating repetitive requests, and providing a standardized interface. Note, however, that the broker, in contrast to simple RFID tags, is reused, so its cost (even high) is not critical.

In the case of location-aware RFID tags, general system functionality is reduced in comparison with simple tags related with pieces. This is due to a fact that unique "serial numbers" of pieces are not tracked. This limitation is important, however, only from the supermarket point of view. For example, an automatic payment system based on collecting RFIDs of goods from the shopping cart cannot be applied. Similarly, any tool for tracking pieces (statistics, internal supermarket management) cannot be used. However, from a client point of view, these features are not of a great importance. Moreover, a mixed solution may be implemented. One start with location-aware tags, and apply simple tags only to the most expensive goods. Once the simple RFID tags are going cheaper, more and more products are equipped with unique tags, and the system functionality grows.

4.2 Network connections

A technical question arises for choosing the most adequate networking standard to contact end-user devices and the local area network of a supermarket. Two radio-communication standards dominated today's market: Bluetooth, and WiFi. Bluetooth piconets are better for reducing local network traffic and automatic detection of devices in range. Automatic interaction of WiFi devices in range is hard to maintain, and overall traffic inside a WiFi cell may be huge, as the cell size grows (WiFi cells provide useful radio signal at a distance of 50-100 meters, in comparison with 10-meter-sized piconets). However, WiFi is more universal, as at the same time it may be used to connect to a local area network – for Bluetooth one needs a specialized router (however, such router may be linked to a shopping cart). The mixed solution, with Bluetooth piconets and WiFi access to a local area network seems to be the best idea. More and more mobile devices are equipped with both interfaces. Thus, we assume that the system (mainly communicating devices in the shopping carts) accepts a network traffic in both standards, and that the clients' devices are equipped with either Bluetooth, or WiFi radio communication (preferably both).

5. BASIC STRATEGY OF USING ACE AGENTS FOR ACTIVE MARKETING

A connection of ACE agents and RFID identifiers makes it possible to change a traditional way of shopping. Usually, the goods are lying on a shop shelf, passively waiting for clients. A client is informed about given goods mainly via previously watched/read advertisements, in TV, newspapers, etc. A real comparison with other similar products is limited, not to say impossible (especially inside a supermarket, at shopping time). We propose to change this situation by introducing an active marketing – "hunting" for clients. Active goods/products are trying to access client's personal communication device in order to present themselves and to persuade the client to buy them. To this goal, RFID tags are used to identify goods and places(locations), and an internal supermarket database of goods is used to collect an on-line information about goods. From the client point of view, the main technical problem lies in intelligent filtering of all incoming offers to restrict such offers only to those interested for the client at the moment. This is a task for client's personal agent, residing in client's personal device and migrating to the supermarket network for requested information. The agent informs its owner about new "interesting" offers. What is "interesting" for a given client is programmed for/by him in the code and variables of his/her personal agents. The agents may also take into consideration such features, as actual client's position, battery time, costs of a connection, etc.

Note that the active marketing is not treated as yet-another version of irritating spam, because all the information coming to the client is filtered by his/her personal agent, and probably the client is waiting for such information. As the client is targeted on "interesting" marketing information, such way of contacts will probably substantially reduce client's individual level of irritation. Moreover, a personal agent will be treated as an advisor rather than yet-again source of useless marketing information. Note also, that the time period between sending a marketing info and buying some goods is very little (seconds or minutes), comparing with traditional marketing in newspapers and TV (a delay counted in days or even weeks).

Personalized interaction of the client's device and different goods makes it also possible to implement individual targeted marketing. A "special" offer sent to a client may be personalized according to (1) the current situation, and (2) a history of contacts with this client. Note that today's "special offers" are not targeted on particular clients except for high-cost, rarely selling goods – a typical "special offer" of a supermarket is addressed to all the supermarket clients, regardless time and money already spent by the client. Our "special offers" may be addressed to individual, however, anonymous (unknown by name/real address, not by phone number) persons.

The above described strategy of active marketing makes it is possible to reverse a traditional way of marketing and shopping. Instead of huge amount of useless marketing information, flooding clients all the day regardless time and place, we propose to search automatically marketing info looking for something "interesting" (from a particular user point of view) at given place and time. We expect at least four scenarios of using the active marketing strategy.

- *Dynamic scenario*: a client stores a list of products he/she wants to buy at the moment. Passing given product from the list, the client is informed by a pop-up message. Then, the client may collect an item of the product to automatically "free" its corresponding position from the list. During searching for products, the client may be guided to certain locations. It is also possible to search for similar products, to compare products according to users' opinions, current price, functionality, etc. Note that no direct interaction is needed except for storing the list – the rest is performed automatically with the help of ACE agents, and RFIDs passing by.
- *Personal Marketing Firewall*: a client stores a list of "needed" and "unwanted" products, to pass/immediately block any marketing information related with these products. Thus, an amount of advertisement may be substantially reduced. The list of "needed"/"unwanted" products may be set up dynamically, similar to the list of IP/DNS sites for Personal Network Firewalls, by answering simple questions once a new advertisement is displayed, e.g. "Accept? Yes/No/Always yes/Always no".
- *Static scenario*: a client chose already a product, however its price (or other conditions) is not satisfactory yet. Then, the client instructs his/her agent to continuously monitor the product/price. Once the price drops below certain limit, the client is immediately informed, for example by an e-mail or SMS message. Note that the monitoring agent resides in the supermarket network for a long time, not only at client's shopping time, waiting for a selected change in supermarket database of products. Note also that such agent is executed autonomously, with no on-line connection with its owner.
- *Hunting for promotions*: a scenario similar to the above static scenario, except the agent is monitoring a group of products rather that a single

product. As soon as a promotion arrives concerning any product from the group, the agent owner is immediately informed.

Note that at least two last scenarios may be implemented with agents working in the Internet rather than in internal network of a supermarket. Thus, an offer of several supermarkets may be compared, and client's choice is significantly extended.

A supermarket may also take some profits, tracking an "interaction" among client's personal agent and different goods. A statistical (anonymous) information obtained in such a way may be used to improve an internal organization of a supermarket (positions of goods, mutual relations among different goods, clients' choices, etc.). Note that this a real, on-line information rather that estimated results obtained by specialized research on chosen clients/situations/goods. Such real immediate information is very important not only for supermarket owners, but also for producers, marketing managers, etc. Among others, a few useful scenarios of using ACE agents to improve an internal organization of a supermarket are given below, both apparent (i.e., with some results visible for clients) and transparent (visible only for the staff):

- tracking client choices,
- tracking client positions and timings,
- computing global statistics concerning the above (and similar) scenarios,
- taking care about some special products, e.g., RFID tags with thermometer/pressure meter, to inform about "bad" conditions while storing and transporting given goods,
- tracking persons with temporary/stable RFIDs, for example children and other family members, tracking positions of shopping carts, etc.,
- tracking and monitoring supermarket staff,
- anti-thief systems,
- automatic counters of shopping carts, clients, goods bought by clients, etc.

6. RELATED WORK

To our best knowledge, a system for supporting everyday shopping in supermarkets has not been proposed so far. Except for some proprietary prototypes, as for example "intelligent shopping cart", "intelligent refrigerator", and some automatic payment systems [7], not a single global solution exists. Moreover, the above mentioned prototypes are not personalized, except for very limited parameterization (the same functionality for everybody). Thus, we think that our idea of active marketing and personalized filtering of advertisement info is pretty new.

Similar, the software agents technology has not been used for shopping support. Typical solutions [2, 14], based on descriptive approach to agent programming [13], are not universal enough. The environments with imperatively programmed agents are "heavy" and less secured [10]. A popular idea of an "intelligent" personal agent [3], introduced a few years ago, is quite different of our approach. First, typical "intelligent" agents are pre-programmed by system designers, and their architecture and functionality is fixed (from an end-user point of view). Second, these agents are not highly mobile, and their code is "thick" - they need a powerful environment to be executed efficiently. As a consequence, a typical personal agent is executed in a stationary, fixed host (usually a private host of an agent owner). Mobile devices, characterized by small memory, display, and keyboard, are not powerful enough to fulfill the agent's expectations. A local area network of a supermarket is also not a good place for such an agent, mainly due to security and compatibility restrictions. In contrast, imperative ACE agents are mobile and distributed, with "light" mobile and "thick" stationary parts. At the same time, private agents may be personalized by their owners, still having a possibility of using other agents, mainly system agents predefined by "trusted" system designers.

A personalized functionality of private ACE agents cannot be implemented using classical programming techniques, mainly WWW servers and centralized, multi-user and multi-access databases. All such techniques assume that there is one single software common for all the users, executed in a certain place and with certain access limits. Even parameterized, such software is not able to satisfy any user. Moreover, a level of parameterization is usually low, due to technical and economical reasons. To effectively filter the marketing information flooding the clients, one needs a specialized, personalized software, adjusted to his/her personal communication device. On the other end, every product is different, and its marketing should be different and addressed to different clients, thus an individualization is needed as well.

Using RFID tags for marketing support was not considered so far. Note that the popular bar codes are not used to this goal as well. Our idea of using RFID tags for broadcasting an automatic info about passing by an "interesting" product/place looks quite new. The technology used is "light" and may be easily applied for today's radio communication systems and existing personal communication devices.

7. CONCLUSIONS

In this paper we present our approach to active marketing in supermarkets, with an extensive use of radio-frequency identifiers (RFIDs) of goods and locations, and personal, mobile communication devices of supermarket clients. Our solution is based on personalized filtering of RDIF-related marketing info by the use of software agents. To this goal, we adopt our Agent Computing Environment ACE framework, previously used for personalizing an access to Web Services and Internet data sources. ACE agents are partially executed in clients' mobile devices, and partially in the local area network of a supermarket. These agents are able to filter broadcasted marketing information according to detailed user needs, and current situation – technical restrictions for end-user device, date and time, previous contacts, etc. Several strategies are proposed to use ACE agents to support everyday shopping in a supermarket, including maintaining shopping lists, personal marketing firewalls, and continuous monitoring and asynchronous notification about changes (promotions, discounts, etc.). Basic functionality of the system may be extended at any time and for any user, for example by enabling contacts with other users, by tracking current locations of persons (e.g., children) and shopping carts, etc. The agents and RFID tags may be also used by supermarket staff in a transparent (from a client point of view) manner, to support internal management of the supermarket.

ACE agents may be executed in an autonomic manner, and it is possible to implement an asynchronous way of information access – this is the agent informing its owner that something interesting just happened. What is "interesting" for a given user is programmed by this user in the agent's code. ACE agents may be executed in arbitrary chosen (by users) places, including stationary network servers and users' private mobile devices. Moreover, a place of an execution of an ACE agent may be dynamically adjusted to current situation/user localization/date and time, etc. As agents may be executed at different places, there is no huge demand for high computational power, neither at server-, nor at the client-side. Due to distribution of network resources and devices, and small sizes of Bluetooth piconets and coverage areas of WiFi access points, an overall network saturation is reasonably small.

The system is flexible and open, not only for new clients and products, but also for new services, communication standards, etc. Due to brokerage of public agents, the new products, services and communication protocols may be added in an invisible (for an ordinary user) way.

Although the ACE framework is fully implemented and industry-tested [10, 11], we could not test the proposed application in a real supermarket. Thus, we developed a simulator, being a set of ACE agents, to model a behavior of a set of RFID tags and different supermarket clients. Even this simplified simulator showed that the system would be very useful, both for supermarket clients, and for supermarket staff. The implementation costs are quite low (a single PC as a server, and a set of Bluetooth/WiFi devices), in comparison with other parts of a local area network of a typical supermarket. We are looking now for a supermarket ready to implement and test our approach with real clients, products, and services.

REFERENCES

[1] Bonett, M., Personalization of Web Services: Opportunities and Challenges, 2001, Ariadne Issue 28, http://www.ariadne.ac.uk /issue28/personalization/intro.html

[2] DARPA Agent Markup Language Homepage, http://www.daml.org/

[3] FIPA Personal Assistant Specification, http://www.fipa.org/specs/fipa00083/XC00083B.html

[4] Franklin S., Graesser A. Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents, 3rd Int. Workshop on Agent Theories, and Architectures, Springer-Verlag, 1996

[5] Milojicic D., Trend Wars - mobile agent applications, IEEE Concurr., 1999, pp. 80-90.

[6] Nwana, H., "Software Agents: an overview", Knowledge Engineering Review 11 (1996) 3, pp. 205-244

[7] RFID Payment Processing Drives Customer Throughput at POS, RFID Technology info, http://www.rfida.com/weblog/2005/01/rfid-payment-processing-drives.ht

[8] Rykowski J., "Agent Technology for Secure Personalized Web Services", 24th International Scientific School ISAT 2003, Szklarska Poręba (Poland); September 2003; pp. 185-193

[9] Rykowski, J., Using software agents to personalize natural-language access to Internet services in a chatterbot manner, 2nd International Conference Language And Technology L&T'05, Poznan, Poland, April 2005, pp. 269-273

[10] Rykowski J., Cellary W., Virtual Web Services - Application of Software Agents to Personalization of Web Services, 6th International Conference on Electronic Commerce ICEC 2004, Delft (The Nether-lands), 2004 ; ACM Publishers; pp. 409-418

[11] Rykowski, J., Juszkiewicz, A., Personalization of Information Delivery by the Use of Agents, IADIS Int. Conference WWW/Internet 2003, Algarve, Portugal, 2003, pp. 1056-1059

[12] Secure Logistics, RFID centre, http://rfid.bemrosebooth.com/secure_logistics.php courier mail

[13] Shoham, Y. Agent-oriented programming. Artificial Intelligence, 60(1):51-92

[14] Specification of the KQML Agent-Communication Language, the DARPA Knowledge Sharing Initiative, External Interfaces Working Group, http://www.cs.umbc.edu/kqml/kqmlspec/spec.html

[15] Thomas D., IT move in store for new Tesco shops, VVU network technical info, http://www.vnunet.com/news/1152559

[16] What is RFID, RFID Journal, http://www.rfidjournal.com/article/articleview/1339/1/129/

[17] Wooldridge, M., Jennings, N.R., "Intelligent agents: theory and practice", Knowledge Engineering Review 10 (1995) 2, pp. 115-152