

Sweeper-Agent Recommendations-Tree Early Scheme

Sameh Abdel-Naby and Paolo Giorgini
Department of Information and Communication technology
University of Trento - ITALY
{sameh, paolo.giorgini} @ dit.unitn.it

ABSTRACT

This paper presents a possible roadmap for what we call a “Sweeper-Agent”: an environment-friendly agents-based mobile technique, which is carried out within the activities of BlueAgents, a newly-formed research project focusing on the integration of mobile Bluetooth communications and Agent-based applications. We foresee an architecture of a service management technique that would increase the efficiency of mobile service. This paper proceeds in four sections. The first section outlines the state of the art of Agent-Oriented Architectures, and the motivation behind our research. The next section explains the Sweeper-Agent framework that uses Agent-Based Recommendations-Tree for mobile application, and how this will help mobile applications developers. Then we introduces Sweeper-Agent foreseeable integrations. And we describes the Sweeper-Agent positioning within the already implemented ToothAgent. Finally, we demonstrates the value-added service opportunities that may occur when Sweeper-Agent is implemented within a mobile-agent architecture.

Keywords: Agent-Oriented Programming, mobile services, Bluetooth communications, ambient intelligence.

1 INTRODUCTION

Ambient Intelligence information systems and mobile-based applications are two promising approaches in the direction of enhancing Service-Oriented Architecture (SOA). Objects-Identification related technologies are encouraging scholars to build up *Smart Ambient* systems that can interact with humans and objects to improve the surrounding atmosphere. The merit goes to *Sensors Technology* in forming a virtual community between environmental objects. This virtual community is reasonably interactive, and it provides data that can be used forward in studying the ambient intelligence theories.

Location-based services help mobile users to achieve daily desires in a shorter and smarter way. A situation in which a mobile user in a certain area is recognized by all of the surrounding devices increases the usability

of a range of resources that may have never been used before. For example, suppose a mobile user in a governmental office is asking for a driving license renewal; the recognition of the surrounding devices in the office space may help this mobile user to be informed that a new legislation is applied and could be clarified at a located office. At the same time, if coordination between agents is set up, another communication can be made to inform the mobile user that the nearby insurance company office would be interested to have a copy of the new driving license.

A previously proposed framework, SpatialAgent [1], aiming at providing services for mobile users based on their locations, utilized a RFID-based location model, *Radio Frequency Identifications*, to identify objects and users locations, and a location-information-servers (LISs) to manage location sensing systems and agent devices. In this system, agents have a graphical interface that allows their interaction with users who can freely customize them. A related research, “follow-me” [2], conducted by Cambridge University’s Sentient Computing Project, assumed that a space is equipped with tracking systems that help the location identification process of users and objects.

Our framework suggests that agents will control the process of mobile devices tracking through maintaining a history records saving method that predicts a range of random locations that a mobile user maybe moving to afterwards. For example, if an agent is able to recognize that the latest communication with a user took place within the university campus, a prediction can be formalized to assume that the user’s next move will be within a specific range of locations such as the parking space or the bus stop. Accordingly, though such a prediction could make the system less reliable, by avoiding the use of the third-party technologies, such as sensors or alternative tracking devices, the framework has the advantage of being technically simpler and cost-effective.

This paper is structured as follows, Section.2 explains the Sweeper-Agent framework that uses Agent-Based Recommendations-Tree for mobile application. Section.3 introduces Sweeper-Agent foreseeable integrations. Section.4 describes the Sweeper-Agent positioning within the already implemented

ToothAgent. Section.5 demonstrates the newly found opportunities after Sweeper-Agent.

2 SWEEPER-AGENT

To explain how Sweeper-Agent technique would work, we imagine the situation of a student mobile-user who is moving within the university campus, from entering to the university, reading in the library to attending a lecture in a classroom. At the same time, we assume that the university has got distributed servers able to recognize the student's moves and make a certain communication with that user. We also assume a that Sweeper-Agent works within a Used-Books offering system, ToothAgent [3], that is able to communicate with mobile users through a Bluetooth connection and exchange useful information corresponding to a student's interests. (Figure.1)

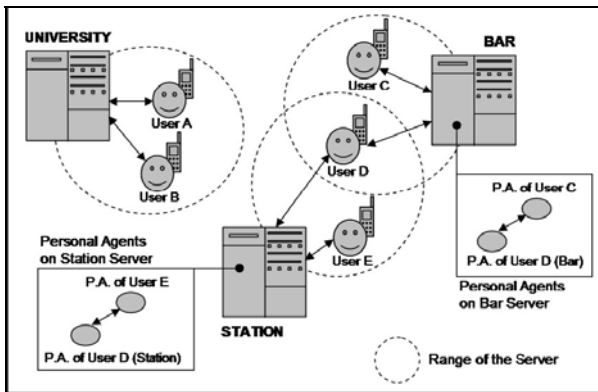


Figure 1: Users, servers, and virtual communities of personal agents. [3]

As long as the student is moving and recognized by these distributed servers, a connection record is always updated and referring to the user's new location. Also, a little but feasible time-to-live (TTL) is added to each updated request, which makes the system obliged to search automatically for the student's new location and update with the new status. Whenever the user is not found within the area that servers cover, a request is passed to the prediction server carrying the Master Sweeper-Agent, in this case - *univ-sweeper-agent*, and therefore a range of predictions is realized and, consequently, the request is forwarded to the nearby connected agents. (Figure.2)

Notably, Sweeper-Agent is to have a hierarchy of agents, as in each area there is a master Sweeper-Agent and a sub-Sweeper-Agent. For example, *univ-sweeper-agent*, that controls the routing of the requests between the university community and the predicting servers; if a student is not located in any of the university areas, this master agent will forward the service request to the predicting servers clarifying the new situation, and the prediction servers decide the "where to go" request path (Figure.3).

A worthwhile risk is taken, as the mobile user may decide to go to a point out of the coverage area, and this will lead the system to a user status recognition failure. Then the mobile-based side of the system will inform the user that a connection can not be established with any of the surrounding servers and, consequently, the service is not available at this point. But this defect could be avoided by identifying a quite wide variety of points that a user would move to after doing a certain action within the coverage areas.

After successfully specifying the location of the mobile user, the service can be offered accordingly. For example, if the user moves from the university campus to the bus stop, and the system succeeds in recognizing this move, a categorized pre-defined services package would automatically be routed to the user. To elaborate, when the user is in the university area, the assigned services offered by the system to this area are offered by the system, used-books selling service and Car-ride sharing possibility. When the user moves to the bus stop, another package of services would be linked to the user profile at this point, such as bus-pass subscription renewal and new bus line schedule.

Categorizing services that could be offered in a certain location is an enhancement that will facilitate the work of mobile-based applications developers. That is because service realization perspective will be taken into account in offering the requested services packages of a specific mobile user without adding any extra developing tools that usually increase the size of the Agent and are time-consuming.

Sweeper-Agents will constitute a tree of pre-defined recommendation paths that a services request (SR) within a network will be following in order to achieve the delegated task. Each of these tree branches will, first, test itself for fulfilling this SR and in case of a negative result, this branch will pass the same SR to the approaching one, and so on until a positive feedback is given by one branch/Agent.

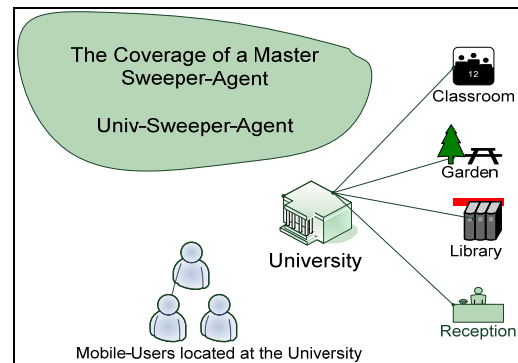


Figure 2: Sweeper-Agent Classification

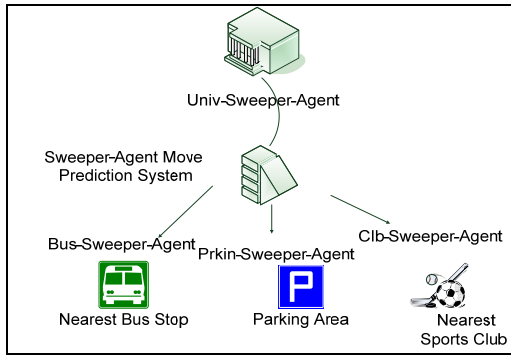


Figure 3: Sweeper-Agent Hierarchy

For a branch to achieve a service request, there are a number of steps that should be taken before the real execution of the SR. For example, once the SR reaches an achieving point, the responsible branch would broadcast its capability to implement this SR, and this broadcasting process would be a bottom-up approach using the same path that the incoming request used. This will help the dedication of the network capacity at the branch level to one request; therefore the result can be carried back faster to the mobile user. Yet, in case the mobile user is out of the coverage area, then the SR will be touring within network nodes endlessly. To solve this problem, we recommend adding a second layer “customized” TTL for each SR to reduce the inappropriate utilization of the traffic, and ensure that the SR result – indicating that the mobile user is out of the coverage area – will be returned to the system.

3 FORESEEABLE INTEGRATION

The ToothAgent [3] architecture is portal multi-agent-platforms-based that can be installed in server where Multi-agent techniques are enabled; each server handles a specific data service area, and system users can contact their personal agents using their Bluetooth integrated devices. Notably, ToothAgent is domain and technology-independent, so specific services provided by a server do not affect the system, and different technologies can be used in different sights.

When a mobile device user gets closer to one of the servers (Figure 1), the pre-installed software on the mobile device sets up a connection with these servers and broadcasts requests related to the accessible services. These requests depend on the previously communicated file between mobile device user and PC. Then agents try to broadcast their requests in the network in search for other agents that carry similar interests or matching criteria requests, and once a fulfilling agent was found, the results are stored locally in the ToothAgent database server. Notably, each of the allocated servers is operating throughout a Multi-agent platform. Consequently, communication among agents and mobile devices is based on one-to-one communication protocol, in a way that allows receiving and processing requests obtained by users. Distinction

between agents is realized through the unique Bluetooth address of the corresponding address [3]. Finally, results are sent back to the user and a list of all visited servers that cooperated with this service is saved to be reviewed by user later on.

Figure 4 shows ToothAgent service accessing architecture. In steps 1-3 the user outline servers and services registered to the ToothAgent Database. In step 4, the user selects the more desirable services and provides the system with relevant information. In step 5, using a Bluetooth, previous data and configurations saved in the configuration file are retrieved by the mobile device. In steps 6-7, the user approaches an architecture-related server, so device software connect to it and start sending service requests upon the available ones. In steps 8-12, a pre-qualification process is initiated to confirm that the user’s interests match the server services and then results are sent back to user. Finally, the mobile device is storing server addresses to record previously interacted servers.

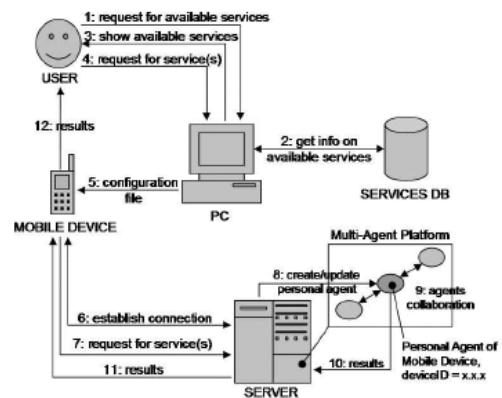


Figure 4: ToothAgent Technique in getting access to service [3]

4 SWEEPER-AGENT POSITIONING

We imagine a scenario where the mobile user X approaches a university campus that is covered by ToothAgent service servers. In the way in, X passes by the university reception, his mobile device will communicate with the location-server and uploading the pre-saved service queries or receiving previously requested data. Afterwards, X moves to the library or the classroom, and yet the location-based different servers sill communicate with X’s “mobile device” and refresh his/her position within the network. Notably, X’s moves are restricted within the Bluetooth coverage range and whenever X leaves the Bluetooth-covered area, the connection with the offered services will be broken.

Nonetheless, one of the previously proposed ways toward the extension of the coverage in such systems is to embed a network of sensors [1] that is able to identify the user’s location and report it to a data-gathering server. These proposed architectures are

quiet effective in mapping the user's moves – using RFID-Tags and Readers (Figure. 5), for example – and, accordingly, provide a service related to his location. At the same time, and apart from the existing sensors-power-consumption problem, the disadvantage of using a network of sensors is the high cost of its embedding and maintenance. Moreover, determining the user's location does not necessarily mean that the system is able to offer him a certain service unless another layer of location-based servers are there to communicate with the user's mobile device and exchange service queries.

Therefore, Sweeper-Agent, in a scenario similar to the one of ToothAgent and taking into consideration the need of to have the above-noted extra layer of servers, will be executed/created whenever the user is not located within a network. In a hierarchal design, the master agent, "Top Agent", will be informed that a user is out of the coverage area whenever a refreshment process (pinging) happens and there is no response from the user's mobile device. The ToothAgent service server will invoke a Sweeper-Agent and send it to the prediction server; the prediction server will first review its issuer name, the *Top-Agent* responsible for the issuance, and accordingly forward the Sweeper-Agent to the pre-defined prediction path.

Once the user is found at any of the predicted path sites, the sweeper-agent will be back again to the ToothAgent service server and a new-place notification process will be launched. Then an old service update will be communicated with the user or a new service can be offered according to the new location. Accordingly, we will avoid the use of new hardware layer, i.e., network of sensors, while achieving the optimum utilization of the network bandwidth. That is because a service package will not be transferred within a single network unless a sweeper-agent reports the successful positioning of a mobile user; otherwise, the sweeper-agent TTL will be invoked and the request will be ignored until further communication with the user.

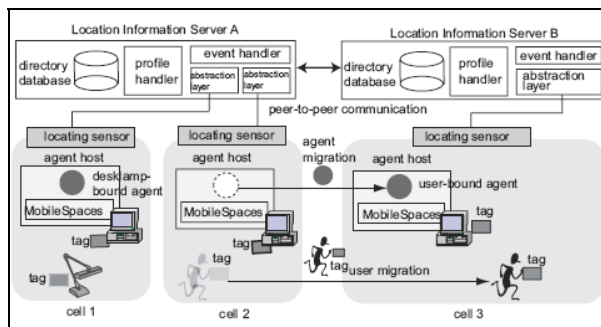


Figure 5: Architecture of the SpatialAgent framework [1]

5 NEW OPPORTUNITIES

Applying Sweeper-Agent in a network of mobile-users will give the possibility for the central system to provide users with services that are not useful unless a user is located at a specific spot and time. For example, a user that is moving from the university and going to the bus stop, and the system has succeeded to identify the user location at the time that another user is moving to the parking area and the system, now, is able to identify both users location perfectly and report their exiting status to the central system, university-sweeper-agent, a possible implementation here is; the link between the two users and matching their destination for Car-ride sharing service.

Car sharing is a method to reduce the usage of cars in a specific town or territory, reducing car usage helps in turn to decrease pollution and prevent some other related problems. This usually takes place by having 1) a car owner who uses his/her car to move from a place to another, and 2) another person who is interested to go somewhere along the car owner's path to destination, and at the same time the *ride seeker* is willing to share the ride cost with the car owner.

That will be achieved easily using a Sweeper-Agent based architecture, as the car owner location will be immediately recognized as a PARKING AREA and the ride seeker location will be immediately recognized as Bus-Stop, therefore, a possible suggestions by the predictions server can be previously stored, which may lead both system users to finally share the trip to a specific destination.

Another similar services can be seen as a new opportunities for providing more feasible benefits for mobile users. For example, a university student that is located by the system at the central university library, down town, and another student that is located in a remote library and looking for book that can be hardly found. A smart matching between the two students can lead the down-town student to deliver the book to the other one at the remote library. Definitely, in such a case, this will require the system to implement user/students profiles in order to realize their relationships with different university campuses, and predict their moves.

FUTURE AND RELATED WORK

ToothAgent [3] is a related work to our research ideas, as we see a convention with the way ToothAgent research team introduced the applicability for a mobile-based application, through a multi-agent framework, to provide mobile users with used-books selling service. Another related work was introduced in March '05, MoPiDig [4], this technique was applied the best within organizations that users-grouping and service-categorization is needed (e.g., museums).

Some of the topics we are planning to focus on in our future research are: (1) Implementing a Sweeper-Agent-enabled version of the ToothAgent, and test it in different situations providing different kinds of services, e.g., car-ride sharing in case the user is by the bus stop; (2) Learning Agents, as the research area that is going to increase the smartness level of agents to understand the mobile users common visiting zones and timings, and accordingly, leading the system to faster predictions; (3) Integrating a smart approach for users to search services in a menu-based style, in which we see a significant potential, as it will give users the possibility to search services linked to other locations for planning or knowledge reasons and finally (4), Agents-Customization technique, introducing the ability for a user to customize the services categories and select only the ones he is interested in.

CONCLUSIONS

This paper introduced the early scheme of Sweeper-Agent, a new technique used for providing a location-based services, this arise when a pre-defined list of locations installed on a prediction server is considered, this list assumes the availability of a mobile-user at specific location after being in another one. We showed that this technique can be implemented within a Multi-agent based architectures in order to facilitate the mission of service packages transfer within a network and ensure a good utilization of the bandwidth. Then we demonstrated the possibility of Sweeper-Agent to integrate with ToothAgent, a already implemented and tested architecture for selling and offering used books among students at university of Trento, Italy. Finally, we illustrated a scenario where RFID technology is used to determine user locations, then we positioned Sweeper-Agent in the same architecture so we avoid the use of any sensors frameworks.

ACKNOWLEDGEMENTS

We first thank the University of Trento - BlueAgents Research Group for the efforts made to carry out a different promising research activities. In particular, we thank ToothAgent research team and ArsLogica s.r.l. for the unabated cooperation and support given to innovative and creative ideas

REFERENCES

- [1] Ichiro Satoh: Mobile Agents for Ambient Intelligence, Proceedings of MMAS 2004: Kyoto, Japan, pp. 187-201.
- [2] A. Harter, A. Hopper, P. Steggeles, A. Ward, and P. Webster: The Anatomy of a Context-Aware Application, Proceedings of Conference on Mobile Computing and Networking (MOBICOM'99), pp. 59-68, ACM Press, 1999.

[3] Bryl, Volha and Giorgini, Paolo and Fante, Stefano (2005) Toothagent: a Multi-Agent System for Virtual Communities Support. Technical Report DIT-05-064, Informatica e Telecomunicazioni, University of Trento.

[4] Seitz, C., Berger, M., Bauer B. "MoPiDiG" , Proceedings of the First International Workshop on Mobile Peer-to-Peer Computing, Orlando, Florida, USA, März 2004