Interoperability of Services in E-Government using Intelligent Agent and Electronic Government Protocol

Assia Tebib, Mahmoud Boufaida
LIRE Laboratory
Mentouri University of Constantine
Constantine, Algeria
{a.tebib,mboufaida}@umc.edu.dz

***Abstract*—The e-government implies interoperability between the various existing administrative services so that the work of institutions can be more transparent for the citizen, more reliable and economic. It is the key problem in the development of the e-government services. To this purpose, we define an approach to develop the mechanism of interoperability of information systems of the e-government. We propose an applicative protocol named EGP (for, Electronic Government Protocol) that permit a technical interoperability by ensuring the exchange and data sharing between Public Administrations. The proposed approach is based on the intelligent agents. A case study is developed to validate the suggested approach.**

Keywords-e-government; interoperability; protocol; intelligent agent

I. INTRODUCTION

The term "e-government" (for, electronic government) generally indicates the exploitation of information and communication technologies for improving the interaction between citizens and government agencies [1]. For instance, e-government systems simplify the citizen's access to services, allow government agencies to effectively deliver their services, and enhance the citizen participation in government and decision-making activities. In the last few years, both local and national governments world wide are planning and realizing e-government programs. These initiatives have been generally welcomed by citizens, and several studies point out that many citizens and businesses require online services from their government [2]. These considerations motivate the enormous, both technological and scientific, efforts performed for improving the range and the quality of the services offered on-line by government agencies.

One of the most interesting research directions consists of defining architectures and frameworks for supporting citizens in their access to e-government services. We can't speak about e-government without approaching the concept of interoperability. It is a key issue in the development of current e-government services. Interoperability among Public Administration (PA) agencies has been identified as a central issue and a critical prerequisite for the effective functioning of contemporary PA systems [3]. Various sources agree on the definition of interoperability considering it as a capacity of various types of networks, computers, operating systems, business applications, software or services (data-processing or not), to work together by using specifications, languages and common protocols, and to give access to their resources in a reciprocal way [2]. The "European Interoperability Framework for Pan-European E-Government Services" (EIF) [4] defines interoperability as the ability of information and communication technology (ICT) systems and of the business processes they support to exchange data and to enable sharing of information and knowledge. EIF defines three interoperability types at the technical level (e.g. among the technical infrastructure of public administrations from different countries), the semantic level (e.g. different meaning and usage of documents, information) and at the organizational level (e.g. among different business processes, goals).

The heterogeneity, distribution, openness, highly dynamic interactions, are some among the key characteristics of another emerging technology, that of intelligent agents and Multi-Agent Systems (MAS). M. Luck et al. [5] propose the following definition: ''an agent is a computer system that is capable of flexible autonomous action in dynamic, unpredictable, typically multi-agent domains".

Intelligent Agents have been extensively applied in the past for handling the distributed management of a wide variety of services (e.g., e-commerce, e-learning, e-recruitment, and so on). Their adoption in the context of e-government, instead, received less attention.

The present paper addresses the issues of technical interoperability by proposing an approach which needs to integrate a component used by intelligent agents for ensuring the exchange and data sharing between the different public administrations. Also, the goal of this paper is to show, by presenting an architecture, that the Intelligent Agent technology not only can be applied, but also can provide important benefits, in the e-government context.

For this, the paper is organised as follows: in section 3, we present an overview of our proposed architecture. In section 4 and 5, we present a specification of our protocol EGP (for, Electronic Government Protocol). In section 6, we provide

some aspects of the approach suggested and in section 7, we present a case study. The final section contains our conclusion and some perspectives.

II. Related Work

Many e-government projects were developed and various approaches were proposed to ensure interoperability. It can be achieved via various mechanisms. One way to achieve interoperability is by only using information systems from a single administration [6]. These systems will most probably be able to perfectly exchange information with each other, by means of a proprietary interface between the systems. But, as the current IT of the public sector is very heterogeneous, and the sector is characterized by a different systems, this is not a realistic solution [7].

Moreover, single-vendor interoperability will not contribute to supplier independence because the use of an proprietary interface by another administration is usually a costly and complex process. Another mechanism to achieve interoperability is the use of open standards [8]. Standards are about collectively agreeing on the specifications for the interfaces between application, services, systems and networks that interact. Open standards differ from proprietary standards because participating in the process of developing, using and maintaining such a standard is in principle open to and freely accessible for everyone. The Dutch government has chosen to prefer open standards above proprietary standard, mainly to achieve interoperability within its IT architecture.

The eGOV project [9] proposes an architecture to enable 'one-stop government', in order to describe services a markup language (GovML) has been developed. GovML defines a set of metadata to describe PA services and life events. The EU-PUBLI.com project [10] defines a Unitary European Network Architecture. It proposes a middleware solution to connect heterogeneous systems of different public administrations and to enable a service-based cooperation between PAs.

The open and standard exchange protocol "Protocole d'Echanges Standard et Ouvert" (PRESTO) [11] specifications, along with clarifications, amendments, and restrictions of those specifications that promote interoperability. Protocols such us eLINK, FAST and ebMS2 are very known.

These projects prove the feasibility of a protocol of data exchange between PAs, while ensuring the aspect of interoperability, but they did not explore the possibility of combining a protocol based on the reference architecture and the intelligent agents.

The specific connotations of the e-government scenarios and the main features of our approach make the exploitation of the Intelligent Agent technology extremely appropriate. Intelligent Agents have been extensively applied in the past for handling the distributed management of a wide variety of services (e.g., e-commerce, e-learning, e-recruitment, and so on). Their adoption in the context of e-government, instead, received less attention. The goal of this paper is to show, by presenting an architecture, that the Intelligent Agent technology not only can be applied, but also can provide important benefits, in the e-government context.

In addition to this, we interesting more particularly in the aspects of interoperability using (the standard in fact) TCP/IP architecture in the distributed environment by integrating a new protocol named EGP (for, Electronic Government Protocol) in the application layer.

This new protocol is used by intelligent agent to ensure their cooperation and interaction.

To our knowledge no work has thought about combining the two technologies: the intelligent agent and the protocol that is integrated into the application layer.

In the following section we present an overview of our architecture which integrates the protocol EGP and the different agents.

III. An Overview of our Architecture

The general architecture of our system is shown in figure 1. From this figure, we can define our architecture by the quadrupled: <UA, PAA, MA, EGP> where:

- User agent (UA), that supports users in their access to e-government services;
- Public Administration Agents (PAA), that support the Public Administration offices in their activities;
- Manager Agents (MA), that uses EGP (Electronic Government Protocol) (detailed in the following sections) to facilitate interoperability with other agents and thus provide good service to citizens.

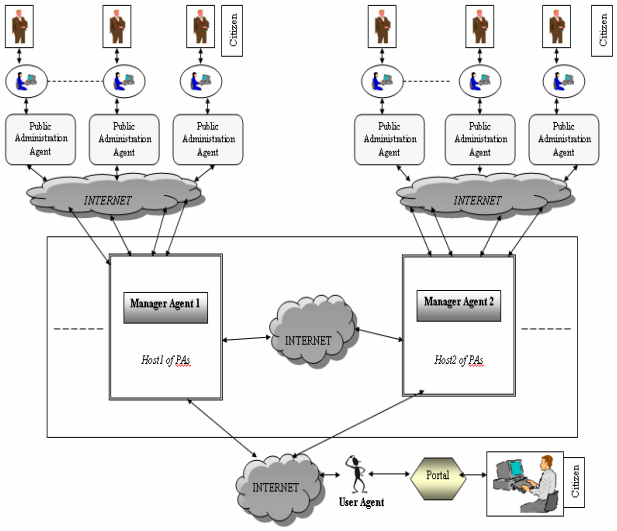

Figure 1. Architecture of our System.

To understand the after, take into consideration that the Public Administration Agents are on the client level, and the Manager Agents are on the server level (see Fig. 1).

According to the presentation of the architecture we define three types of interaction:

- An interaction between User Agent (UA) and Manager Agent (MA) by exchanging messages.

- An interaction between Public Administration Agent (PAA) and MA.
- Interoperability between the different Manager Agents (MAs) using the protocol EGP witch will be detailed in the following sections.

In the following sections we describe the protocol EGP (Electronic Government Protocol).

IV. SPECIFICATION OF THE EGP PROTOCOL

A protocol is a set of rules and it is characterized by a mechanism associated with it. The EGP is integrated at the PA (for, Public Administration) level and the host level. The PA may use the database server associated with it. This connects all administrations of the same type, in other words, it centralizes in one database. The different hosts can interoper together to provide a response to citizen. To understand its operation, we will first give the format of a query and its treatment and the EGP response. So, the mechanism of interaction between the PA level and the host level, and interoperability between several hosts are defined.

A. Parts of the EGP Request

There are two parts in the EGP request: the command and the header. The command corresponds to the various commands that the protocol employs to dialogue (see table 1). They are sent to the host concerned and are interpreted by this last. The second parts, is the header which consists of several lines whose form is always the same one. The name of field followed by sign of equality (=) and the value that each one wants to associate with him.

TABLE I. DIFFERENT COMMAND OF EGP

Commands	description
CONNECT	Provides authentication of the PA from the server to which it was connected.
GET(*)	Find all information about a citizen from his number.
GET(Firstname,Secondname)	Can find the full name of the citizen from his number.
GET(Firstname)	Provides the first name of the citizen from his number.
GET(Secondname)	Can find the second name of the citizen from his number.
GET(Blood-group)	Get the blood group of the citizen.

The fields can be various types according to the associated use. Its value is that one which was seized by the civil servant of the PA counter. The table 1, presents only a few commands defined by the EGP.

B. Parts of the EGP Response

The EGP answer consists of several of the same lines format. Each line corresponds to a field that is required by the request of the client, follow-up of the sign (=) and of the value of the field which is in the data base of the associated host.

V. INTERACTION BASED EGP BETWEEN PA AND HOST AND THE EFFECTIVE INTEROPERABILITY

The interaction between the client and the server is the first communication mode of the EGP protocol.
The client must send a request near the host that is associated for it. The command part of the EGP request can be one of the commands used by EGP for communication (see table 1).
The communication between the client and the server is established in two phases according to the mechanism of the EGP. The first step consists of the authentication of the public administration near the host. The second step carries out the interaction between the client and the host by sending the request established by the PA with the host concerned. (The two phases of the EGP protocol have been already explained in [12]).

Before sending the answer to the client, EGP host will use two characters which are specific for him and which facilitate the interaction between the client and the host. One is character (# 0) (a sharp followed by number 0) and the other is character (#1) (a sharp followed by number 1).
We will use the sequence diagram of UML to show the mechanism.

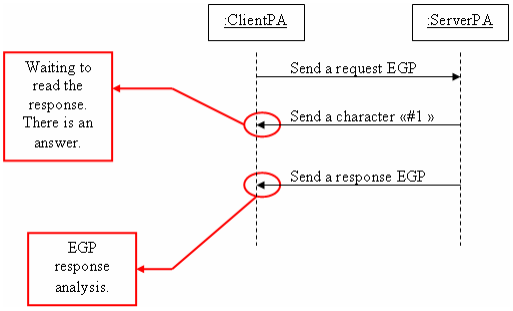

Figure 2. The case of existence of the EGP response.

The particularity of the EGP protocol is that ensure the interoperability between several hosts.
The distributed hosts owe interoper to provide a service and a help to the citizen. However, this communication must ensure the interoperability in order to obtain the communication between hosts.
When a client sends a request which cannot be satisfied by the host to which it is connected, the communication between the hosts is established (interoperability).
The processes of the technical interoperability in this case is consists of two phases: the first one is the authentication of the host near to another, and the second one, consists of the effective EGP interoperability by sending a request.

A. AUTHENTICATION OF THE HOST

We can connect to the host of the government, starting from address IP and of port 3366 of the EGP protocol.
Once connected, the host can be authenticated near to several hosts, by using the same format of EGP request of authentication employed in the case of client/server.
Once authenticated, the host will be able to send its request to the other hosts near of which it was authenticated to

supplement missing information by interoperating and to post then on the interface of the PA.

B. *EFFECTIF EGP INTEROPERABILITY*

The particularity of the EGP protocol is that ensure the interoperability between several hosts.

The exchange of commands and characters (#0) and (#1) emphasized the concept of the interoperability.

The commands exchanged between the hosts are the same ones in the case client/server.

The use of the characters (#1) and (#0) remains always valid in the dialogue between the hosts to ensure the interoperability. In the case where is an answer after treatment of the request, the host sends a character (#1) to warn the host receiving that an answer exists and that it will be sent. In the contrary case, a character (#0) is sent to it. Also, certain commands will be refuted by the host which appears by the sending of a character (#0) to the requester host.

The client sends a request to the host by using the EGP client (Fig. 3). The request is received by the component of EGP host 1 (admitting that this request is sent to host1) what corresponds to the step 2.

This last seeks information in its data base in substitute the command by the adequate request (step 3). When the information required by the client exceeds those of the host 1, it transmits a request which supplements missing information with the EGP client 1 which constitutes a component of EGP in the same host (step 5). This part adopts the behavior of a EGP client. In other term it sends the request to the component

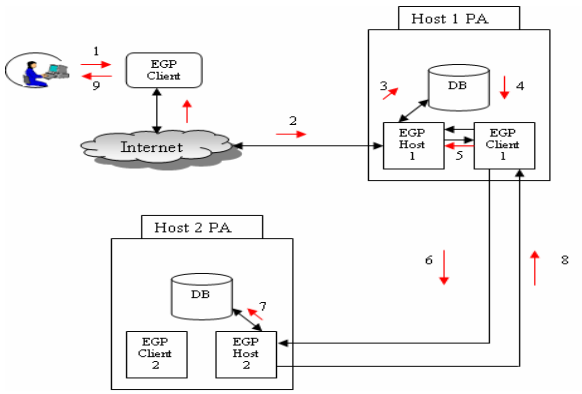

Figure 3. Interoperability between hosts.

EGP host 2 (step 6) that carries out information retrieval corresponding to the command sent. According to the planning of Fig. 3, we notice well that the behavior in distributed environment is the same mechanism as in the client/server model what corresponding to architecture P2P (for, Peer to Peer); and which characterized our protocol.

Once the found answer, it will be automatically sent to the EGP client 1 (communication of the type request/answer) (steps 7, 8). This last, sends it directly to the EGP host 1 component.

The EGP client receives the answer which will be then posted on the interface of the agency of the public administration what corresponds at steps 5 and 9.

VI. DIFFERENT COMPOMENTS OF OUR ARCHITECTURE

We have defined our architecture by the quadrupled: <UA, PAA, MA, EGP>. After the specification of our EGP protocol, we will now highlight the different components of intelligent agents (User Agent "UA", Public Administration Agent "PAA" and Manager Agent "MA") used in our architecture.

In the following sub-sections we describe in details all the different components of our architecture.

A. *The Public Administration Agent*

Each Public administration (PA) is managed by an agent. It manipulates the EGP (Electronic Government Protocol) to facilitate interoperability and thus meet the needs of citizens.

When a citizen presents himself to a counter for a given service, the PA activates the Public Administration Agent. This last, builds its query using the EGP protocol in order to obtain information about the citizen by exchanging the data between agents. In this case, the EGP protocol of the PAA collects all information necessary by sending the request (previously defined format) to the Manager Agent of the public administration concerned.

It should be known that the agencies (physical counter) of the PAs that belong to the same branch of industry are all connected (centralized) to the same host. Consequently, they share the same data base. The used approach is similar to the architecture of bus communication JAVA-RMI (Remote Method Invocation). In this architecture, there are the Skeleton host side and the Stub client side [13].

B. *The Manager Agent*

The Manager Agent (for, MA) uses EGP to facilitate interoperability with other agents and thus provide a good service to citizens. For this, each Manager Agent is placed on a host. This component (MA) consists of the EGP server, the EGP client and the database (Fig. 4).

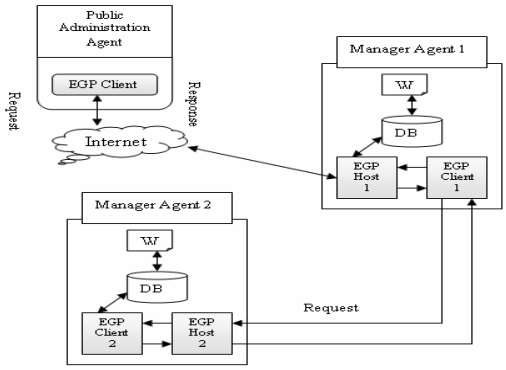

Figure 4. Architecture of the manager agent

The EGP (host protocol side) used by the MA1 receives the request and then seeks in its data base.

If the response is available in its data-base it will send it to the requester counter (PAA). If the information which is needed exceeds those which exist in its data base, the EGP

(host side) of the Manager Agent should interoper with others hosts that already exist and will transmit the request.

The use of characters (#1) and (#0) remains always valid defined in the mechanism of the EGP protocol.

C. *The User Agent*

The citizen can directly reach the portal of the government via Internet and can select the service which he needs. The type of selected service enables him to be connected directly with the host of the administration that offers this help. We should take into consideration that HTTP protocol is present in our architecture and in which all the advantages remain available. We will use it in the exchange of the web page. Owing to the fact, that it is regarded as a standard, it will facilitate the installation of interoperability.
A User Agent (UA) is associated with a citizen. Its support data structure stores the profile UP (User Profile) of the citizen. UA is activated by citizen when he wants to know if there exist new services interesting for him and, in the affirmative case, to access them (Fig. 5). The user agent uses two databases: the service database and profile database.

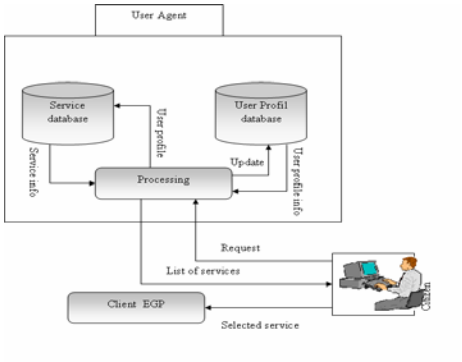

Figure 1. The component user agent.

The profile database is constituted of the threefold <Idui, Di, Ksi> where :

- Idui : is the identifier of the citizen;
- Di: is the geographic information;
- Ksi: is the set of keywords.

This last is composed of two parts : <Mci,Coefi> where :

- Mci : is the set of the keywords already used;
- Coefi: is a coefficient calculated with the Jaccard method (Jaccard Coefficient) [14].

First, a user agent retrieves UP from User Profile Database and stores it in its support data structure. Then, it allows citizen to send a query Qi, consisting of a set of keywords specifying the services he presently desires.

After this, it sends the pair <Qi, UPi> to the component "Prossessing", which is in charge of processing the query and constructing a list SLi of services possibly satisfying a citizen.

The component "Prossessing" (Fig. 5) constructs this list and proposes it to citizen; it can select the most relevant services, according to his needs.

On the basis of his choices, "prossessing" suitably updates UPi. Such an activity consists of two steps, namely:

- Insertion/Update of interests. In this case a user agent examines both Qi and the features of the services of SLi chosen by user. It adds into Ksi each keyword/ feature not already presents therein, and associates a suitable, initial interest degree with it. In addition, for each keyword/feature already present into Ksi, it suitably updates the corresponding interest degree.
- Pruning of old interests. In this case UAi removes from Ksi all keywords appearing non longer interesting for citizen on the basis of their interest degree. The pruning activity allows UAi to maintain UPi up-to-date and, at the same time, avoids the excessive growth of the size of UPi.

Finally, when citizen ends his activities, User Agent inserts the updated UPi in the user profile database. Once the service selected by Citizen, EGP protocol (Client EGP) allows the user to be connected directly to the server that provided this service by selecting the right Manger Agent which can interact with other MA.

VII. CASE STUDY : "MANAGEMENT OF PAS : HOSPITALS, COMMUNES, POLICE FORCES"

We illustrate our architecture via an application of a suggested scenario; the end-user can be a citizen or a PA. We define three types of agents: an agent that manages the different communes, an agent of the various hospitals, like that of various police forces. Each one of these agents centralizes a whole of PAs that are associated for it and can interoper.
In our suggested application, the citizen presents himself at the level of the hospital (physical counter) for a given service (such as his admission to the hospital). The counter related to a hospital identifies the citizen from his number which is in his electronic identity card. This number is searched.
The EGP protocol client side sends an EGP request by using the command GET(*) which is associated with the number of the citizen. All the information contained in the database of a hospital is accessible by this counter which represents the interaction between the PAA and MA.
On the other hand, the commune cannot reach certain data contained in the data bases of the other MA relative to this same citizen.
The counter of a hospital, automatically sends two commands using the EGP protocol, GET(*) to the MA of the hospital to have all information in the data base of the hospital and GET(Firstname,Secondname) with that of the commune, so the interoperability between the commune and hospital agents appear.
Moreover, there were a communication between PAA/MA and interoperability between two MA, that of the hospital and the commune.

VIII. Conclusion and perspectives

The concept of interoperability is usually present when we speak about installation of the e-government. The present paper addresses the issues of technical interoperability by proposing an approach which integrates intelligent agents ensuring the exchange and data sharing between the public administrations (interoperability). We set up an applicative protocol named EGP (for, Electronic Government Protocol). This last was added by the application layer of architecture TCP/IP. EGP is used in a distributed environment whose commands employed in the communications are simple. This facilitates the data exchange between the PAs. In this protocol, we have specified two types of communication.

The EGP protocol will be used by different agents of the e-government for ensuring interoperability. We have presented a multi-agent system for supporting public administration to provide good service to citizens.

We also identified three types of agents (PAA, MA, UA) that exploit the EGP.

We have pointed out that the proposed system is an attempt of applying the Intelligent Agent technology in the context of e-government. With regard to this, we have shown that many features typical of agents can play a key role in this context.

Our next work will consist in harmoniously supplementing the public portal of the e-government which provides services to the citizens being anywhere in the world. We will also integrate ontology for capturing more concepts about the e-government domain and life events. Further life event and services descriptions will be integrated into the portal and a real one stop portal will be developed.

References

[1] C. Avgerou, “Discourses on innovation and development in information systems in developing countries research”. In Proceedings of the 10th International Conference on Social Implications of Computers in Developing Countries. Dubai School of Government, International Federation for Information Processing, 2009.

[2] G. Marchionini, H. Samet, and L. Brandt, L, “Introduction to the special issue on Digital Government”. In Communication of the ACM, pp. 24–27, 2003.

[3] M. C. Andrew, “E-Government Leadership Rhetoric vs. reality—closing the gap”. Technical report, http://cio.ny.gov/press/News/092110EGovRpt.htm , 2010.

[4] IDABC, “European Interoperability Framework for Pan-European e-Government services”, Version 1.0, In. European Communities, Luxembourg, 2004.

[5] M. Luck, P. McBurney, O. Shehory, S. Willmott. “The AgentLink Community: Agent Technology: computing as interaction - A roadmap for Agent-Based Computing”. AgentLink III; 2005.

[6] Bureau Forum Standardization (BFS), http://www.open-standaarden.nl, 2010.

[7] V. V. Fomin, M. K. Pedersen, H. J. de Vries, “Open standards and government policy: results of a Delphi Survey”. Communications of the Association for Information Systems, 459–484, 2008.

[8] K. Zhu, K. L. Kraemer, V. Gurbaxani, S. X. Xu. “Migration to open-standard interorganizational systems: Network Effects, Switching Costs, and Path Dependency”. MIS Quarterly, 2006.

[9] The EGOV project, http://www.egov-project.org, 2004.

[10] TheEU-PUBLI.comProject, http://www.eupubli.com, 2004.

[11] PRESTO, « PRESTO, un protocole d’échange pour les besoins de l’administration éléctronique », 2007.

[12] A. Tebib, M. Boufaida, “Une Architecture pour l’Interopérabilité et la Sécurité des Services de l’E-Gouvernement », CRiSIS’2007 (the International Conference on Risks and Security of Internet and Systems). Marrakesh, Morocco, 26—28, 2007.

[13] Vercouter, “Intergiciels : Java Remote Method Invocation”, website, http://www.emse.fr/~vercouter/cours/intergiciels/rmi.pdf., 2010.

[14] Wikipedia contributors, “Jaccard index”, Wikipedia, The Free Encyclopedia, 2011.