Smart E-Service Implementation as mobile Agent in a Smart E-Government Platform

Sinan Adnan Diwan  
Collage of Computer Science and Mathematics, Wasit University, Wasit, Kut, Iraq.  
Limkokwing University of Creative Technology, Selangor, Cyberjaya, Malaysia.

Dr. Sundresan Perumal  
Malaysian Islamic University, Information Technology Collage, Malaysia.

Dr. Dhyaa Shaheed Siber  
Collage of Computer Science and Mathematics, Wasit University, Wasit, Kut, Iraq.

Abstract
This paper introduces the concept of Smart-E-Service to be the kernel building block of the Smart e-Government. The presented Smart-E-Service has many privileges over the traditional E-Service such as fully cross-platform, social, liable, negotiable, autonomous and mobile. Smart-E-Service is implemented in this paper as a deliverable mobile agent that complies with HTML5 as a frontend and Node.js modules to interface JADE platform at the backend. The presented Smart-E-Service is preemptive behavior rather than reactive or even proactive; preemptive interpreted as actions based upon hard domain intelligence. HTML5 terminologies such as Real Multi-Threading and Web Sockets have been exploited and deployed to efficiently increase the performance of Smart-E-Service and sustain its attributes.

Keywords: Semantic Web, Personalization, Personalization Techniques, Extensible Markup Language

Introduction
E-Government is defined as the employment of the Internet and the world-wide-web for delivering governmental information and services to the citizens [1]. The kernel core of E-Government is the E-Service which it is what citizens expect from the E-Government to deliver (i.e., provide/publish).

E-Services designed to be consumed per requested citizen; this happened in two scenarios: 1-Deliver E-Service entirely to citizen side and consume it there; this approach imposes delivering E-Service as a standalone application; this has to consider the compatibility of the delivered E-Service module with the designation platform. Many serious challenges face this approach such as security risks, enough knowledge to install/uninstall E-Service, and experience to troubleshoot malfunctioning of installed E-Service; this is what ‘Apps store’ from Google and other providers are doing.

2-Decompose E-Service into backend component to be consumed on the server side, and frontend component to be delivered and consumed at the citizen side; this approach throws less burden on the citizen in term of starting the E-Service but it leaves the delivered E-Service with less functionalities and having the same issue of compatibility issue. Both presented scenarios do not respond to the specification of Smart-E-Government, which is the next generation of E-Government.

In Smart-E-Government, E-Services hold more smart behavior in its interaction with citizens and the environment.

E-Service Vs. smart-E-Service
E-Government is a collection of E-Services published for the public citizens as part of the commitments of a government. E-Service can be defined in a broad sense as the provisioning of services over electronic network [2]. The production of E-Service considers three main components to produce reliable E-Services [1, 2]:

- E-Service Implementation and Environment: E-Service should be implemented to be cross-platform consumable software module; this is crucial in E-Government due to...
the huge size of population. E-Service implementation should be adaptive to the environment holds be citizens to be consumed properly.

- E-Service Publishing and Discovery: E-Government can be considered to be a large repository of E-Services. UDDI (Universal Description and Discovery Integration) is a software
- E-Service Delivery and Integration: this issue draw main schemes of accessibilities to E-Services published by the E-Government where citizens can invoke and consume available E-Services.

Smart-E-Service is an E-Service delivered from E-Government repository to citizens’ machine to fulfill their requests by autonomously contact information resources and retrieve knowledge.

Smart-E-Service encompasses the ability to perceive the environment and cognitively adapt to the rapid changing within the deploying environment to fulfill citizen needs.

The ultimate objectivity of Smart-E-Service is to help citizens making decisions in certain problem domain by acquiring knowledge to make that decision more effective; Markman in [3] introduced the effective thinking in problem solving as high quality knowledge where problem solving is less about a flash of intuitive brilliance and more about the application of knowledge [3]. Smart actions are produced by software modules and applications installed in a device, for a software to be smart it should be able to assist clients making wise decisions by building semantic network over the domain and developing knowledge by traversing that network autonomously. Knowledge development is accomplished without the intervention of clients.

E-Service Delivery Scenarios

E-Service is a software module when consumed it produces valuable effects on hosting environment, traditionally there are two approaches to have E-Service delivered to designation system: first is the direct installation of the software service, here the service is written using platform dependent languages (e.g., C++, J2ME, Special Java for Android); this service is published in global repositories which client should login and download it. The second approach is accomplished by exploiting web browser where E-Service is written using web languages like XML and JavaScript; this way E-Service can be delivered anywhere to the destination system since there is web browser installed (i.e., some Web technologies like HTML5 provides new tags that need updated versions of internet explorer).

E-Service Liability

E-Service is a software entity that consumed over the clients’ machine by exploiting available resources; this raises many alerts and risks of exposing personal and sensitive information, especially when considering Smartphone that holds very sensitive information about the client/citizen. In this paper we define E-Service feasibility as a set of beliefs initiated and developed at the run time: \( th \) belief, and belief is a set of ontologies that represent initial restriction concepts (i.e., what are authorized actions and responses).

Intelligent Software Agent and JADE

Intelligent Software Agent is a software entity that can perceive the environment through its sensors and acts on that perceiving autonomously. The perceiving is domain specific and a sophisticated process that needs ontological representation for the concepts compasses the environment [4]. In Agent based system, the problem domain is represented using ontology which is defined as the following:

\[ Ontology = C, O, R, P \]

Where

- C: are concepts defined over the problem domain
- O: Object within the problem domain
- R: Relationships between concepts
- P: properties of Concepts

JADE is a developing environment written in Java languages to assist developing Agent systems.

In [4] JADE is presented in details, thus in this paper we are not going to introduce JADE and focus on building wrapper classes only.

Smartphone

There are many definitions for the meaning of Smartphone but the most convenient shared definition is: “Smartphone is a device with high level of computation power and large screen” [5].

Smartphones are devices combined both the functionalities of regular phone and other functionalities which were only implemented on different devices such as the PDA, PC or Laptop. Smartphones significance came from the operating system (e.g., android from Google and windows mobile from Microsoft); the operating system provides enough resources to lunch applications that can collect knowledge for the user of the Smartphone; this imposes the availability to connect information resources such as the internet or information repositories[3][6]. Smartphones are running the latest versions of web explorers such as Chrome, Firefox, Internet Explorer and others.

HTML5 –The New Platform for Developing Software

HTML5 is a co-operation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). New technologies have been introduced by HTML5 like Web Socket and worker Multi-threading; these new technologies area promising environment for developing more complex and safer software modules; this is due to the limited access right granted to the web browser which is the shell under which HTML5 based software modules are executed[7].

Enterprise web applications are tend to use web browser as Thin client; this is where all needed to execute web applications functionalities are available in the web browser, and what is missing can be added as plugins (e.g., flash player). Most of today’s smartphones have web browsers that
support HTML5, CSS3 and JavaScript which represent the only requirement to develop more sophisticated software [7].

**Google V8 engine and Server Side JavaScript (node.js)**

V8 JavaScript engine is an open source JavaScript engine developed by Google in Denmark and shipping with the Google Chrome browser. V8 adds a virtual JavaScript machine and a special operating system (i.e., Chrome OS) which runs on x86 machines and Advanced RISC Machines (ARM) [8]. V8 increases performance by compiling JavaScript to native machine code before executing it, rather than to bytecode or interpreting it. Further performance increases were achieved by employing optimization techniques such as inline caching [8, 9]. Node.js is an open source toolkit for developing server-side applications based on the V8 JavaScript engine. Node.js is written in C++ which is the same language used to write V8 engine; this brings a lot of correlations among programs written for server-side and client-side, anyway. Node.js extends JavaScript API to build web applications at higher level of scalability and extreme performance due to the non-stop feature [9].

V8 enables C++ applications to expose its own objects and functions to JavaScript where objects are wrapped in V8 templates. Two types of templates are available: Function template and Object template, and C++ object’s methods are accessed in an internal field stored within function template instances. [9]

---

**Smart-E-Service implementation as Mobile Agent (The Proposal)**

The proposed system is a collection of software technologies, methodologies and protocols to implement E-Service as mobile agent module; this has been a worthwhile effort to promote the deployment environment toward introducing smart and later on intelligent behaviors.

We have objectivities to be fulfilled in the proposed system as the following:

1- Implementing E-Service as smart, social, mobile and autonomous software module.

2- Implementing E-Service as fully platform independent software module that works on Smartphones, laptops and other devices that have Chrome or Firefox internet explorer.

3- Implementing E-Service as on demand utility that can invoked without a need for installation.

4- Implementing E-Service with attributes that grants E-Service the ability to exploit and develop domain intelligence.

As it has been introduced in previous sections of this paper, web browser has been evolving to be a rich development environment for software applications; this is especially recognized with the emerging of HTML5 tags (i.e., Multi-threading, primitive data types); we focused on developing E-Service as JavaScript module which is transferred to requested client and launched in the internet explorer. Using JavaScript was due the promotion of this language to be client side as well server side (i.e., Node.js is a server side JavaScript environment).

Implementing E-Service as JavaScript supports the fulfillment of objectivities (2) and (3); this JavaScript E-Service is designed to be FIPA-Compliant or in other words, to be compatible with JADE platform as in figure (2), and this approach supports the fulfillment of objectivities (1) and (4):

As figure (2) presents, E-Service is delivered to the requested client and behave as JADE agent; this has been a challenge due to the need to equip JavaScript E-Service with a lot of functionalities (i.e., JADE is a collection of classes used by Agent to accomplish its tasks); these are so hard to be implemented all using only JavaScript, thus we have implemented it as two components: first one is a light client side JavaScript module which is to be delivered over the internet to the client and Second one is heavy server side JavaScript module which is designed and implemented as Node.js module called JADE-Wrapper library, as it is presented in figure (3) JADE-Wrapper library also contains many interfaces to JADE utilities (i.e., RMA agent, Sniffing agent and others).
Figure (4) presents the conceptual view to the essential components that compose the proposed System.

Algorithm (1) is the backend Java Agent which is designed and implemented using JADE environment.

Figure (5) presents the implementation partial Node.js code JADE wrapper module with the conceptual view. Figure (5) presents partial HTML5 code to load and start Smart-E-Service.

Algorithm 1: Backend Java Agent Interaction to JavaScript Mobile Agent

```
public class AgentProxy extends Agent {
    public void setup() {
        InstantiationException RequestQueue as new ConcurrentLinkedQueue<String>();
        InstantiationException ResponseQueue as new ConcurrentLinkedQueue<String>();
        Instantiate TempQueue as new ConcurrentLinkedQueue<String>();
        Instantiate new CyclicBehaviour();
        public void action() {
            if (reqQueue.size() > 0)
                while (reqQueue.size() > 0)
                    Begin
                        var mobileAgentMsg = RequestQueue.poll();
                        var msg = ACLMessage.Translate(mobileAgentMsg);
                        TempQueue.push(msg);
                    End;
                End;
                addBehaviour(new CyclicBehaviour());
                Begin
                    while (msg.size() > 0)
                        var str = GetResponse(String(msg2));
                        ResponseQueue.push(str);
                    End;
                End;
            End;
        }
    }
}
```

Figure (7) presents a screenshot of starting Node.js module in a Virtual Private Server (VPS), till this research Node.js does not have JADE wrapper despite the fact that JADE is a widespread Agent development environment.
Figure (8) presents JADE GUI has been started with ProxyAgent and a Mobile Agent instance in figure (8) JavaScriptMobileAgent01_Agent is launched due a call from the client side over http connection or WebSocket connection. Figure (9) presents sniffing utility been deployed to capture ACLMessages exchanged between ProxyAgent at the backend and the Mobile AgentE-Service at the frontend.

Figure (10) presents ACLMessages exchanged between Smart-E-Service and Java Agent running at the server side.

Conclusion

1- JavaScript running under V8 engine introduces efficient open source development environment in developing large applications. Along the implementation of the Mobile Agent using JavaScript language, many JavaScript engines have been tested (i.e., Firefox, Internet Explorer, Safari, Opera), and V8 Chrome shown recognized performance level.

2- Implementing Smart-E-Service as JavaScript mobile agent has added significant attributes to the E-Service like autonomous, social and other features mentioned in above sections. Embedding intelligent software terminologies within enterprise web application is expanding the opportunities of having smarter enterprise software modules.

3- JavaScript is an open source language; this introduces new challenge in implementing E-Service using JavaScript where client, with little experience in JavaScript, can hack E-Service manually or automatically, in the proposed system we divided the processing into frontend and backend; this methodology hardens the penetration into the E-Service.

4- Smart-E-Service mobility is valid only during the web session, when requester closes the session Smart-E-Service shuts down all its behaviors (e.g., interacting the environment, socializing other agents and mobility); this issue has cons and pros where from a side it increase code safe execution; this due to the limited authorization of web browser and from another side it affects the availability of the Agent.

5- E-Service Mobility is guaranteed when implementing Smart-E-Service using JavaScript language; this is due to the standard terminologies adopted by all smart devices (Smartphone, Smart PDA, ..., and others) that have updated web browser installed. SmartE-Service can be viewed as a web service that runs on Node.js, and maintain all invocation and deploying standards such as XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), HTTP (Hyper Text Transfer Protocol) and WebSocket.

6- Implementing Smart-E-Service using JavaScript increases the potential of Web application to hold Intelligent Infrastructure of the next generation of E-Service, which is the Intelligent-E-Service
References


