ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

The SandS Ecosystem, a True Instance of WEB4.0

Apolloni B[1], Mlakar I, Lipaj A, Galliani GL

Dept. of Computer Science, University of Milan, Italy

ABSTRACT: We introduce a peculiar ecosystem aimed at ruling in remote the household appliances of the members of a special social network. The keen feature of the social network is a networked intelligence, equipped with cognitive tools that enable it to provide services fully compliant with the members' needs. The scheme is the following: The appliances are internet-connected through the home Wi-Fi router. The user asks the social network for a task to be executed by his appliance (for instance, washing three kilos of woollen coloured laundry), the network, in the role of an electronic super-mom, sends directly to the washing machine an optimal sequence of commands the recipes (such as: warm the water at 34°, soak for 57 minutes, etc.) to execute the task in a way that matches the user preferences, possibly green goals included. Feedbacks are sent by user and appliances themselves to the network intelligence to close the permanent recipe optimization loop, with offline advice on the part of appliance manufacturers. A properly devised user interface allows a friendly and accurate management of all interactions between the user and the social network, constituting the user-centric support of the cognitive driven services representing a genuine instance of WEB 4.0.

KEYWORDS: Networked Intelligence, Internet of things, Cognitive Tools, Cognitive Web Services, Social Appliances, Mqtt Protocol, Mqtt supported IoT

I. INTRODUCTION

In spite of the many ambiguities and contradictions in the definitions of the WEB levels, we refer to the sharp graph posted by Nova Spivak1 in Figure 1. Both the 3.0 and 4.0 releases of the web encapsulate artificial intelligence tools. Inthe former the tools are based on classical logic rules based on Ontologies (Knowledge Graphs in the new notation [1] and Semantic knowledge representations [2]. In addition, the 4.0 issue uses cognitive tools, like the Fuzzy Inference Systems [3] and Reinforcement Learning [4]. WEB 3.0 is aimed at matching the services delivered with the true intention of the requester, as contained, possibly implicitly, in his request. WEB 4.0 is aimed at matching the delivered services with the intention of the requester as emerging from both the current request and the past service delivering history on this requester, jointly with his complete profile (thus containing both environment information, such as his house and the number of children, and information on his personal habits, such as his green mood, taboos, etc.).

In this way, web services become really user-centric, and our ecosystem is a concrete instance of the new deal of the web. With some abuse, we may say that the cognitive functions of the system we introduce are enabled by its networked architecture, which in turn may be considered as an extension of the neural network in the brain of the user, so that the information processing milieu is the same at micro scale (the user brain) and at the macro scale (the Internet of Things). It is a rather hierarchical architecture that may be nicely resumed by the virtual super-mom metaphor in Figure 2. It hides a rather complex system, as a result of the European FP7 Project Social and Smart SandS that in these pages we consider from a few specific aspects that will head next sections: information architecture, web intelligence, and user interfaces and communication protocols. Each of them deserves some advance in respect with the state of the art, which we will resume in the concluding section.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

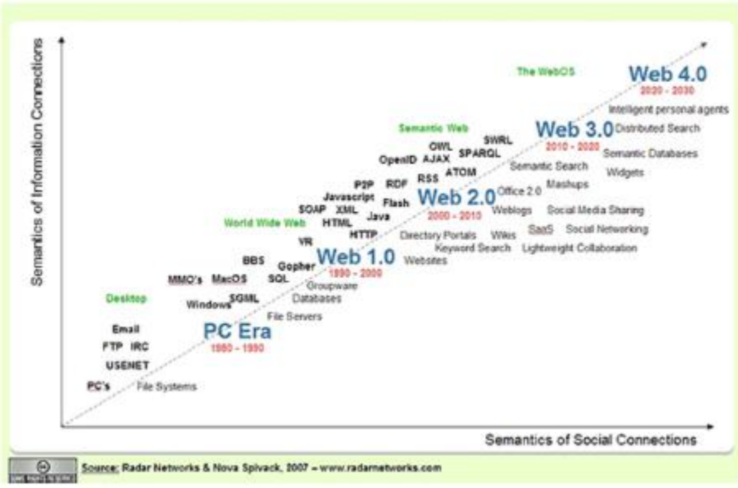

Figure1: TheevolutionoftheWEBdeliveries

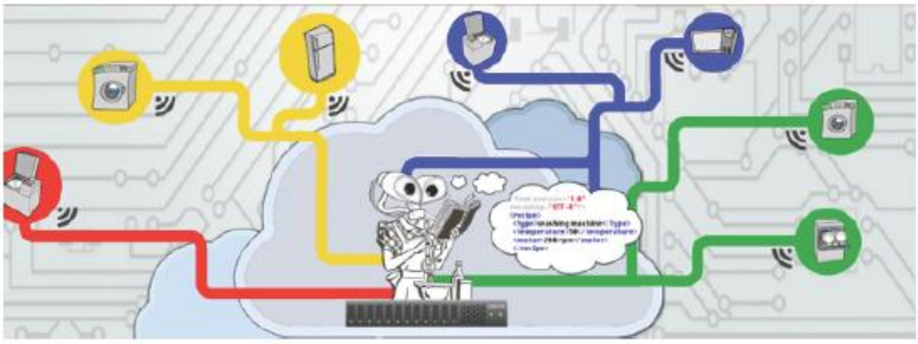

Figure2: TheSandSSuper-Mom

II. THE INFORMATION ARCHITECTURE

Like with the neural network in our brain, the cognitive functions highly depend on the connection network. This is realized through the layered architecture shown in Figure 3, which replays the levels of the information processing, and an information flow like in Figure 4, which pipes data in a complex loop.

Namely, on the top we have a very peculiar social network of facts (ESN), whose members are both users (eahoukers, from a funny compression of easy house-workers) and appliances who pipe their feedbacks to the network, as a share capital, and receive recipes as a service. In the middle the Domestic infrastructure (DI) which is the hub of all information flows. DI makes the recipes compliant with the home rules. On the bottom the appliances, interfaced to the network through an electronic board.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

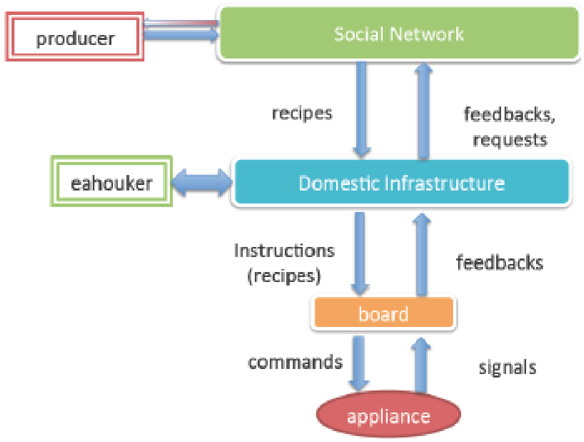

Figure3: TheSandSmainstream

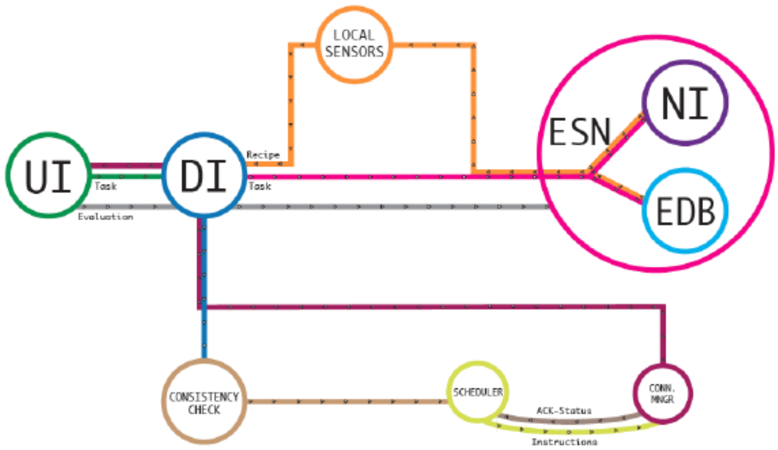

Figure4: TheSandSdataflow

While the two directions of the mainstream are clear – Top-Down: recipes produced by ESN, feed instruction by instruction by the DI to the board and translated in machine commands by the latter; Bottom-Up: signals interpreted as feedback by the board, forwarded by the latter to the ESN via the DI – the overall sync of the data trafficking is rather complex. Namely, the metro-map of the information flow, shown in Figure 4, passes through three main stations: the social network (ESN), the user interface (UI), and the connection manager (CM), a DI module which collects messages from or to appliances and one hub: the DI as awhole. The ESN-UI loop vehicles tasks requests and feedbacks in one direction, recipes in the other. The ESN-CM loop conveys recipes plus local sensor signals to the DI. Along the way, they are transformed into appliance instructions fed to the CM. The return way transfers appliance and user ACKs till the ESN.

II. THE COGNITIVE WEB INTELLIGENCE

Like in the human brain and in the complex societies as well, intelligence stands for learning, where learning, in turn, is routed in an information network.We use specially devised cognitive algorithms to optimize recipes on the basis of the information triplet (task, recipe, feedback), where the task is issued by the user (e.g. concerning a laundry), the recipe is a list of commands generated by the network to be dispatched directly to the appliance – the core business of the ecosystem (e.g. “close the door”, “charge water one half capacity”, “heat 34 degrees”, etc.), and the feedback is produced by both user and appliance w.r.t. the recipe execution – the true social contribution to the network life (e.g.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

"softness good", "cleanness poor", etc). Each task is parameterized into a definite, appliance-dependent set of features, which are used at the ground level of recipes selection. We have two levels of adaptation of the recipe to the user.

- At central level thanks to the availability of the feedbacks, the true trigger of any cognitive system
- At local level through the implementation of the "home rules", in terms of Horn clauses having in input environment conditions, such as noise or the house electrical fee, and in output some restrictions on the recipe execution.

The scenario is the following: a huge set of information concerning the user's behaviour, as a static part, and their actions, condensed in the information carrier (task-request, recipe and evaluation). This information is logically connected by space-temporal links and physically connected to the intelligence of the network by the Internet network gathering users and appliances. We use a divide and conquer strategy that splits the goal of producing optimal recipes that are compliant with the user preferences into three serial tasks Figure 5.

1. Mining: Identify similar recipes within the EDB. Here similarity is measured, through proper metrics, in regard of both the request content and the requester profile (ranging from his house location, personal preferences, etc.). If you find a request very similar to the current one, then use the corresponding recipe. Otherwise, pick the five most similar requests and use the corresponding recipes as starting point of the subsequent task
2. FIS (Fuzzy inference system). Use a particular FIS algorithm (that we discuss here below) to infer the best recipe in response to the request. When a huge set of (recipes, evaluations) has been collected, enter the third task
3. RL (Reinforcement Learning). Use the above specific dataset to statistically finetune the recipe via usual Qlearning algorithms.

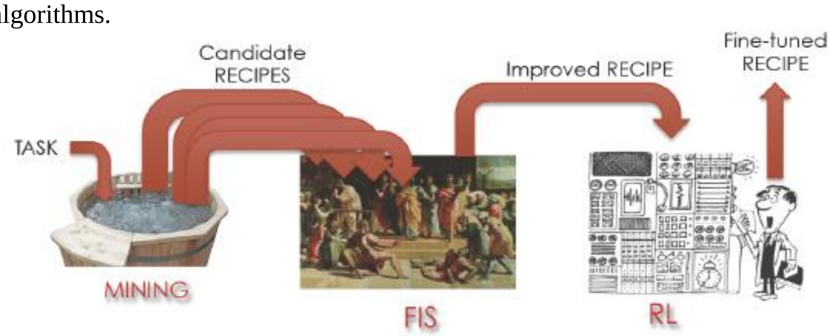

Figure5: The three-steps procedure to produce a recipe

The principal trait of the networked intelligence job is that the information carriers are produced by the humans in their extremes (task-request and evaluation) and must be synthesized into definite recipes (the central segment of the carrier) by the network intelligence. The latter simply consists of a series of learning algorithms having the goal of satisfying the user, which results in a fuzzy target. The contrast between:

- The fuzziness of the information produced by the human and the learning goal and
- The accuracy requested to the recipe, is rather unprecedented. This pushed us to introduce a new learning paradigm that we call "learning from nowhere". Technically it is a Sugeno fuzzy rule system where, in addition, both the universe of the discourse and the relation between the output of the Sugeno functions and the parameters on which the user expresses its evaluation are unknown.

For instance, in the case of a recipe for producing a loaf of bread through a bread-maker we face peculiar fuzzy rules such as the following:

- If the loaf is less crusty and soggy then increase rising time
- If the loaf is very crusty and crisp then decrease rising time

These rules may be easily framed in the Sugeno reasoning model where the consequents are crisp variables to be computed through a weighted mixture of functions depending directly on the input variables. In formulas:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

$$y = \sum_{i=1}^{k} \overline{w_i} f_i(x) = \frac{\sum_{i=1}^{k} w_i f_i(x)}{\sum_{i=1}^{k} w_i}$$

Where fi defines the activation function (for short Sugeno function) of the i-th rule, whose shape and arguments depend on the chosen model, wi denotes the satisfaction degree of the premise of the rule, and x refers to the crisp input variables. The distinguishing features of the above rules come from the fact that when the user says that “the loaf is very crusty", s/he has no reference metric variable to set to a specific value, even though s/he distinguishes different kinds of crustiness that gathers in the same quantifier “very crusty”. This denotes that variables xjs exist, but they are hidden to the eahouker her or himself. Nevertheless, to identify a suitable value of y we must estimate their values. A second difficulty is connected to fi in two respects. On the one hand we hazard functional forms such as linear and quadratic ones to interpret the consequent “increase or decrease” of the rising time. But this is a common plague of the Sugeno approach. On the other hand, we don't know the relation between the operational parameter (the rising time) and its effect on the crustiness and humidity appreciation on the part of the eahouker. This entails a non-trivial identification problem to be faced, in a way reminiscent of distal learning [5] in Neuro-control with a two-phase (identification and control) algorithm.

This kind of systems learns fast. For instance, we trained it on a batch of 120 examples like in Table 1 and got a FIS parameterization that is daily used in our laboratory to produce the bread and permanently refined on the basis of the corresponding evaluations. The working cycle consists of 5 phases (first leavening, second leavening pre-cooking, cooking and browning), each specified by a duration and a temperature. The produced bread is evaluated according to 4 criteria in a Likert {-5,…,+5} scale.

FIRST LEAVING		SECOND LEAVING		PRE COOKING		COOKING		BROWNING		EVALUATION			
Time	Temp-arature	Time	Tempe-rature	Time	Tempa-rature	Time	Tempa-rature	Time	Tempa-rature	Frangrance	Softness	Baking	Crust
1821	33	2788	35	575	59	3885	118	101	126	+3	+1	+1	+1
1792	33	2813	35	569	59	3872	118	100	126	0	+2	+2	+2
1820	33	2787	35	570	59	3885	118	97	126	0	+1	+1	+1
1766	33	2754	34	560	59	3889	117	93	124	0	+2	+2	+2

Table 1: An excerpt of the bred maker training set

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

IV.THE USER INTERFACE

The User Interface was designed to enable the user to have a complete overview and control of the status of his domestic infrastructure. The functional goal was to deliver services emerging from: the context of user's current intention, user's complete profile and user's service. The Front End (FE) manages the engagement with appliances and recipes, and the management of user profiles and of the entire household infrastructure.
The tasks are issued in natural language as either a speech (using Google's Speech API2 and Pocketsphinx.js3) or a written document. In order to extract the base parameters (e.g. type of appliance, base features of the task) of the recipe we adopt POS-based approach [6] combined with grammar and n-grams [7].
Extended services allow the user to preview and modify his preferences, and to have a complete overview of the status of his domestic infrastructure. The complete overview also includes the mentioned home rules, a functionality through which each user can incorporate his personal habits as well as his perspective on how the domestic environment should function in terms of resource consumption (water, electricity), noise generation and some abuse cost of resources.
Architecturally the interface FE was developed as a stand-alone web server. It is an Apache server delivering instances dynamically generated by HTML 5.0 documents incorporating context specific JavaScript and CSS. It also incorporates Ajax technology through which user requests are mediated to the web core. In turn, the web core is connected to SandS network via the DI and a set of bi directional web-services hosted by FE server and DI. As outlined in Figure 6, the communication between users, FE server and DI is fully bi-directional. The communication in direction FE→DI is always triggered by user and implemented via a java-script-based triggering of Ajax events, which are transformed into HTTPS-based JSON requests. The received JSON responses are then translated into appropriate UI elements and displayed to the user. The communication in DI→FE direction can be triggered by any of the SandS layers. It can involve notification, procedures for future user's engagement or even a direct UI control. Namely, each message sent to the UI is pre-processed by a JavaScript-based client that identifies the context of the message and dynamically loads or generates appropriate HTML elements or triggers further JavaScript events. This means that actions and reactions may be triggered either by the user (Ajax +) or through messaging service, by any of lower Sands layers (e.g. appliances, DI, etc.).

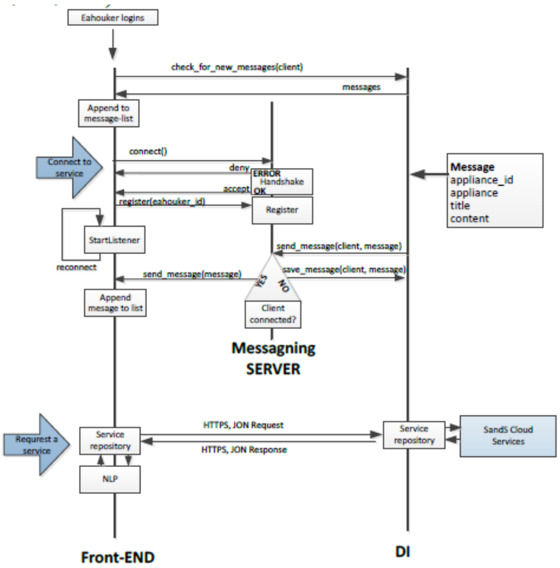

Figure6: Functional outline of communication protocol for delivery of user centric interfaces and services

The SandS system can signal, through the developed bi-directional concept, the user not only with useful information but also with future involvement required from the user and thus fully supports the Web 4.0 philosophy. The messaging API are based on WebSocket Protocol v134 and supported by most of mobile/desktop browsers (tests were made on Firefox, Chrome and IE 8.0+).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Vol. 3, Issue 10, October 2015

V. MQTT PROTOCOL

The underlying communication technology required special effort as well. In order to have an infrastructure that proves transparent to the user, we decided connecting appliances and users to the network via Wi-Fi, whilst the two top layers of the architecture are in the cloud. This deserved us two specific problems: security and communication bi-directionality. With the former we must be sure that our neighbouring doesn't decide burning our thanks giving turkey or, even worse, damaging our appliance. Hence a TLS (Transport Layer Security) layer has been stated on the appliance-DI communications and analogous expedients on eahouker-DI communications. Two-ways communication became a must for IoT instances, which allows event driven protocols with a notably safe of transmission bandwidth. Rather than via a keep-alive mechanism, messaging between DI and appliances is carried out just when one of the two extremes of the communication channel needs transmitting data to the other extreme. An example of a Recipe execution work-flow is shown in Figure 7. Namely, the two-ways communication having both DI and appliances in the twice role of client and server is guaranteed via the adoption of the libraries of a MQTT server5, articulating the communication through four kind of transacted messages (besides service messaging such as PING):

A. CONNECT: the connection request sent from the client, containing QoS setting
B. CONNACK: the server response to a client CONNECT message
C. PUBLISH: once client establishes a connection to DI-CM, both client and server may send a message composed of a topic (the Key to identify the information channel), and a payload (specific data)
D. PUBACK: the response to a PUBLISH message

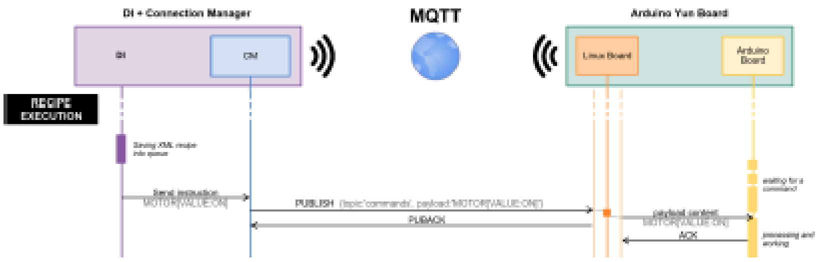

Figure 7: Recipe execution work-flow

To communicate with the DI, the appliance needs some additional intelligence as well, to interpret the instructions into command in the one direction, and the signals into feedbacks in the other direction. This has been realized on an Arduino developing board6 whose functions may be quickly embedded – in the case of the adoption of the SandS paradigm by the manufacturers – into slight extensions of the actual microcontrollers of the appliances.

VI.EVALUATION AND RESULTS

Experiments were carried out, in the PlanetLab facilities7, in order to evaluate the large scale dimensioning and scalability of SandS system through standard software performance testing procedures. The test procedures involved load and stress testing, soak testing and spike testing. The outline of the experimental setup is given in Figure 8. Therein, we used 3 dislocated SandS servers for hosting SandS services and 36 dislocated client machines (servers). In experiments the clients simulated up to 20.000 user-requests per experiment (up to 300 requests per second) and there were up to 40000 virtual appliances connected to SandS system. SandS components and the system as whole were tested using iMacros clients simulating user-behaviour as well as python clients simulating a set of requests to the more exposed SandS services. During the experiments we collected data regarding system load, service response times and number of messages being exchanged.
The most important aspect of the large scale experiments was to identify the correlation between service delivery times and request intensity, as specifically outlined in Figure 9 in correspondence of the main services. The performance of these services mostly exhibits the dependency on the availability of system resources rather than network resources. As such, SandS primarily exhibits features of load scalability. In general it can easily expand and contract its resource pool to accommodate heavier or lighter loads. The current hardware configuration of MIDI/OVH servers could in long-term

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

support 5 FE servers (2100 users), or a demand generated by 9600 users constantly requesting for HMI services of the SandS system (prolonged exposure, non-peak traffic). In terms peak traffic (short bursts) it could support demand generated by up to 14800 users (4 FE servers). As for geographic scalability, through experiments executed via dislocated architecture and via PlanetLab nodes located all over Europe, we observe that SandS system maintains performance and usability regardless of concentration in a local area or a more distributed geographic pattern.

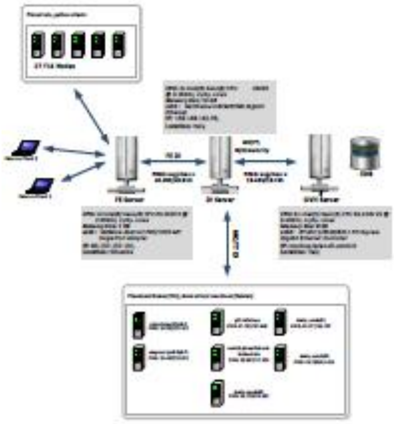

Figure 8: SandS experimental setup at One Lab

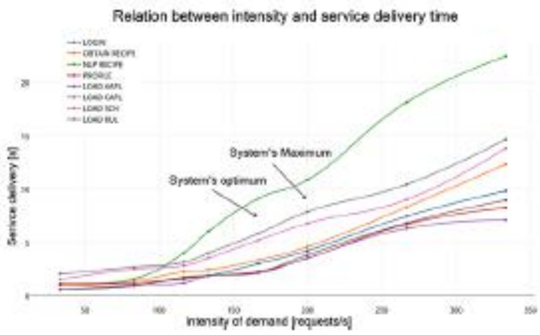

Figure 9: Course of services average delivery time at client's end points

VII. CONCLUSION

In this paper we have outlined a novel eco system which is aimed at managing the household appliances remotely through a social network. In addition to architectural and functional aspects of the SandS platform, we have also outlined a series of large scale experiments which were used to evaluate its dimensioning and scalability. Keen features of the proposed system are a networked intelligence, equipped with cognitive capabilities, and the based service-oriented architecture of the system.

Most of related state-of-the art systems consider smart homes and ambient assisted living (AAL) as a whole rather than a particular sub-domain such as smart kitchens and smart kitchen appliances. For instance in Smart Kitchen forAmbient Assisted Living [8] a novel design of smart kitchen incorporating AAL services is introduced. The system is based on Open Services Gateway initiative (OSGi)8 and AAL Open Association (AALOA)9.

The ground base of the system is the physical context correlated with the environment via logical rules and decision processes. In addition to white goods, the system integrates standard sensors (gas, fire, smoke, flooding) in order to detect emergency situations and automatically take evasive action procedures. In [9] authors represent a system which goes beyond traditional scope of user modelling (and context processing). User's individual preferences are derived from his "Current Lifestyle" and "Future Lifestyle" that play a prominent role in the final decisions. Another platform considering smart home as a whole is a platform represented in [10]. The system incorporates a distributed network

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

which can, through a knowledge base (set of logical rules and correlation) and a smart appliance library anticipate needs of users and react to them accordingly and in cooperation with different appliances. The system establishes a finite vocabulary of activities correlating smart appliances (and their capabilities) and user's tasks via a rule generation incorporating user's behaviour. In CloudFridge a cloud based platform is represented which supports user-experience-centric interaction with an internet refrigerator based on appliance usage incorporating its current and past states. A completely user-centric platform, focused especially on modality of user interfaces, is Kochbot [11,12] which integrates speech interaction into a networked kitchen that can be controlled via textual recipe instructions or spoken commands.

The system we propose overrides all these experiences in that:

1. The ground base of Sands is a cloud located social context allowing the recipe, as the final outcome of a regular transaction, to be specified by a set of cognitive algorithms based on fuzzy rules, which exploit:
    a. The feedback delivered by both users and appliances
    b. The complete history of the user and his profile and the social similarity with other users maintained in the social network database.
2. The physical context of domestic environment is incorporated through home-rules which summarize user's habits as well as the condition of his domestic environment.
3. Natural language processing as well as speech recognition is tools of this platform as well. They are used in order to deliver a speech-assisted intelligent kitchen that can be beneficial to cooking beginners as well as people with disabilities.

Furthermore, SandS platform has also been proven to be both vertically and horizontally scalable, thus exhibiting load as well as geographic scalability. Furthermore, due to its modularity and functional architecture the SandS system could be extended to support any appliance or physical context (sensors and actuators) of a smart home, thus providing a basis for a novel IoT platform for smart homes as well as AAL.

REFERENCES

1. Mitra P, Wiederhold G, Kersten M, "A Graph-Oriented Model for Articulation of Ontology Interdependencies", Springer Berlin Heidelberg, pp. 86-100, 2000.
2. Corby O, Kuntz DR, Gandon F, Zucker FC, "Searching the Semantic Web: Approximate Query Processing based on Ontologies", Intelligent Systems, IEEE, 21: 20-27, 2006.
3. Nauck D, Klawonn F, Kruse R, "Foundations of Neuro-Fuzzy systems", John Wiley & Sons, Inc. 1997.
4. Grondman I, Busoniu L, Lopes GA, Babuska R, "A survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients", Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, 42: 1291-1307, 2012.
5. Jordan MI, Rumelhart DE, "Forward Models: Supervised Learning with a Distal Teacher", Cognitive Science, 16: 307-354, 1992.
6. Rojc M, "Web-based Architecture RES based on Finite-State Machines for Distributed Evaluation and Development of Speech Synthesis Systems", J. of Computational Linguistics Research Volume, 2: 1, 2011.
7. Socher R, Bauer J, Manning CD, Ng AY, "Parsing with Compositional Vector Grammars", In Proceedings of the ACL conference, 2013.
8. Blasco R, Marco A., Casas R, Cirujano D, Picking R, "A Smart Kitchen for Ambient Assisted Living", Sensors, 14: 1629-1653, 2014.
9. Heidari M, Allameh E, Vries DB, Timmermans H, Mozaffar F, "Decision Modelling in Smart Home Design", In Ambient Assisted Living and Daily Activities, pp. 211-218, 2014.
10. Son H, Tegelund B, Kim T, Lee D, Hyun SJ, Lim J, Lee H, "A Distributed Middleware for a Smart Home with Autonomous Appliances", In Computer Software and Applications Conference (COMPSAC), IEEE 39th Annual 2: 23-32, IEEE, 2015.
11. Alexandersson J, Schafer U, Rekrut M, Arnold F, Reifers S, "Kochbot in the Intelligent Kitchen-Speech-enabled Assistance and Cooking Control in a Smart Home", In AAL-Kongress, January 2015.
12. Vadivel R, Bhaskaran, "V. M Energy Efficient with Secured Reliable Routing Protocol (eesrrp) for Mobile Ad hoc Networks", Procedia Technology, 4: 703-707, 2012.