Easing Communication Means Selection Using Context Information and Semantic Technologies

Koldo Zabaleta, Pablo Curiel, and Ana B. Lago

Deusto Institute of Technology, DeustoTech
MORElab, Envisioning Future Internet
University of Deusto, Avda. Universidades 24, 48007, Bilbao, Spain
{koldo.zabaleta,pcuriel,anabelen.lago}@deusto.es

Abstract. Due to the advances in the ICTs in recent years, the variety of communication means at our disposal has noticeably increased. Some communication means offer simple tools to inform users about the status and availability of their contacts. However, not all of them offer this functionality, and indeed, elements like context, preferences and habits of the contact, which currently are not considered, should be taken into account before trying to stablish communication with a contact. In order to tackle this problem, we propose a platform that takes into account these features to assist users in deciding what communication means to use with each contact in each moment. Considering the real-time constraints of the system, a series of performance tests are also presented.

Keywords: semantics, mobile technology, communications, social networks.

1 Introduction

Due to the advances made in the ICTs in recent times, the variety of communication means at our disposal has noticeably increased. Latest smartphones, rather than providing users with just voice call and short message-based communication, enable them to keep in touch with their contacts using most of the communication means available nowadays. Taking into account the remarkable market share increase of these devices, a great number of mobile users have access to several communication means at their fingertips and thus, whenever they want to talk to their contacts, have to decide which to use among them.

This decision is not straightforward, as many variables must be taken into account. For this purpose, some communication means offer tools which inform users about their contacts' availability. For instance, many instant messaging systems have a user state which allows users to know if a contact is online, busy or absent, offline or its time of last connection. Similarly, some social networks give users the option to establish a custom state where they can give out what they are doing. However, other elements like context, preferences and habits of the contact determine the success or failure of a communication attempt. And currently no communication tools which consider these variables exist. Therefore, in many

G. Urzaiz et al. (Eds.): UCAmI 2013, LNCS 8276, pp. 231–238, 2013.

cases the person initiating the communication has to follow a trial-and-error approach, performing several attempts or trying with different communication means before successfully getting in touch with a contact.

In order to address this problem, in this paper we propose a platform that assists users in deciding what communication means to use with each contact in each moment. This platform enables users to define diverse use preferences about their communication means (which can be either general use preferences, like liking or disliking a particular communication means, or context-based preferences, like not using a specific communication means at certain place or time). Using these preferences, when users want to talk to one of their contacts, the system recommends them the most appropriate communication means to use for this purpose. This recommendation has real-time constraints, as users would hardly find it useful if they had to wait for too long before receiving the suggestion. Thus, to determine if response times comply with this mentioned requirement, a performance analysis of the system is also presented.

The remaining of the paper is structured as follows. The next section discusses related work. Section 3 gives a generic description of the platform. Section 4 describes tests carried out in order to assess the performance of the communication means recommendation mechanism. Finally, in Section 5, conclusions and future work are exposed.

2 Related Work

The Semantic Web is an idea promoted by Tim Berners-Lee [3]. The objective pursued with this idea is to add semantic content to the web. With this addition it is possible to share, reuse and process with different applications the data available on the web.

Ontologies, which are used to model the different domains of a software solution, are one of the main tools of the Semantic Web. To create an ontology a knowledge representation language is needed. Nowadays Web Ontology Language (OWL) [16] is the standardized and recommended language by the W3C.

Thanks to the work carried out in this field in recent years, there are several reference ontologies which can be used to represent a specific domain. An example of this type of ontology is FOAF [6], which is used for modelling information about people and relationships among them. Many projects that need to model users make use of this ontology [1] [7] [13]. SIOC [5] is other relevant ontology, designed for representing concepts related to social networks, and which is used in several projects that model this domain [1] [7] [10].

These ontologies can be perfectly used for modelling the areas for which they were designed. The problem arises when there is a need to model an environment that spans more concepts than a single ontology supports, as it is the case of the working domain of this article. Thus, the reviewed ontologies model certain subdomains, such as users and social networks, but we could not find any ontology for representing all the communication means considered and the preferences to define over them, not at least with the minimum details that are required.

For this reason we designed an ad-hoc ontology which adds and extends terms from existing ontologies and vocabularies such as FOAF, WGS84 [15] and OWL-Time [14].

On the other hand, one of the key features that ontologies provide is reasoning over the knowledge represented using them, enabling applications to perform personalized recommendation of products and services. For instance, Rung-Ching Chen et al. [4] present an application that models characteristics of diabetic patients, and which through reasoning about this information is capable of recommending the user which drug to take. A similar approach is followed in our work, which, after modelling the communication means of the users and their preferences over them, reasons about this information to suggest users which communication means to use at each moment.

On to other matters, Knittel et al. [9], carried out different studies in which they determine the most appropriate type of context information to use in order to identify users' status. And Tang et al. [12] developed a prototype which enables users to share context information with their contacts. Although these proposals deal with context information as a means to ease interpersonal communication, neither of them provide the caller with a recommendation about which communication means to use in order to stablish a communication with a callee.

3 The System Architecture

With this work we try to solve the problem of deciding which communications channel to select when a mobile user wants to communicate with a contact. In this section we describe the platform we propose for recommending users the best communication means to use in each situation. This platform is composed of a mobile application and of a central server. These two components are described in detail below.

3.1 The Server

The server is in charge of storing all the information related to the users of the platform, their communication means and the preferences defined over them. The server has two main components, which are the ontological model used to represent the mentioned information, and a rule engine, which processes this information to generate the recommendation about which communication means to use in each situation.

The Ontology. To model the elements that are part of the solution developed we designed an OWL ontology. This ontology models three main concepts.

First, we have the user modelling. Currently there are several ontologies created for modelling user profiles, among which FOAF is the one that has received more support. For this reason FOAF has been used as the basis for user modelling in the proposed platform.

The second main modelled concept are the communication means, represented as a class hierarchy to favour extension and reuse. This part of the ontology is entirely of our own design and does not use any external element. We have modelled the majority of the communication means available on a smartphone as a class hierarchy.

The third concept modelled in the ontology are the communication preferences defined by the users. These can be of general purpose (e.g., "I like telephone calls") or linked to the user's context (e.g., "At work I do not use instant messaging" or "Weekends I do not check my work email account") and allow users to express which communication means are the best to get in touch with them in each particular moment, and which are less likely to be appropriate or successful.

The Rule Engine. Rules are a useful and simple tool to generate additional information from a model. In our case, a rule engine is used to rate communication means based on user context and their preferences. And based on this rating, a sorted list of communication means is obtained, from the most appropriate to get in touch with a contact to the least appropriate one.

Rules are divided into four different groups. The first group identifies whether a user has network connection in the moment of its execution. The second rule group rates communication means according to the user having or not network connection. The third rule group rates communication means which provide users' status (e.g., instant messaging services) according to that kind of information. And the last group rates communication means based on the preferences associated with them.

3.2 Mobile Application

The mobile application is the client of the platform and acts as interface for the users to access its functionalities. Due to its ubiquitous nature, the access it provides to most of the communication means available nowadays and its ability to provide rich context information, it is the perfect device for this task.

Thus, this application enables users access and configure its profile in the platform, registering the communication means they have access to and the use preferences they want to define over them. It also provides a personalized contact agenda, which requests the server the best communication means to use to talk to each contact when the user wants to do so. Finally, in order to enable this recommendation, the mobile application is responsible for periodically reporting to the central server user-specific context information (location, status of the connection, user status in those communication means that provide that kind of information, etc.).

4 Evaluation

In this section we describe the different types of tests we have carried out in order to evaluate the server performance regarding the communication means recommendation. This functionality is the most computationally demanding among the ones exposed by the server, as involves executing a rule engine and semantic inference tasks. At the same time, this functionality has real-time constraints, as it is requested by the mobile application when the user wants to talk to a contact and users would hardly find it useful if they had to wait for too long before receiving the suggestion. In order to evaluate this response time from the end-users' perspective, we have defined the user experience levels shown in Table 1, which are based in the study by Shaikh et al. [11].

Table 1. User experience depending on server response time

Time (t)	User Experience
$t < 1$ s	Very satisfactory
$1 s < t < 3 s$	Satisfactory
$3 s < t < 5 s$	Acceptable
$5 s < t < 8 s$	Poor
$t > 8$ s	Not Acceptable

Therefore, a series of tests were carried out to assess the performance of this recommender service under diverse situations and with different configuration parameters. In the first test case we made requests to the server from the mobile application via Wi-Fi, 3G and 2G networks. In the second test scenario concurrent calls with a different number of clients were made to the server. The third test scenario consisted in making calls to the server with a different number of users registered in the system. And the fourth and last test case consisted in requesting the communication means recommendation for users who have defined a different number of preferences. Except for the third test scenario, we registered two users in the system. And excluding the last test scenario, each user had four preferences defined.

For these tests, the server was executing on a virtual machine. The host server was an Intel Xeon Quad-Core at 2.83 GHz and with 12 GiB of memory, running an Ubuntu Server 10.04.4 LTS 64 bits edition and with a VirtualBox 3.0.14 installation. The virtual machine had the 4 cores and 2 GiB of memory assigned and was running an Ubuntu Server 12.04 LTS 64 bits edition. The web service was deployed in a Tomcat 6.0 using JDK 1.6. Regarding the client side, two different configurations where used. For the network types test scenario a Samsung Galaxy Nexus with Android 4.2 was used with an ad-hoc version of the mobile application which registered and saved the request times to a CSV file. For the other three test cases Apache Bench [2] was used from a laptop computer through Ethernet connection. This enabled simulating scenarios of up to 64 concurrent clients without network being a bottleneck, and also having a more stable network connection which minimises the added noise to the conclusions of these three tests. JConsole [8] utility provided by the JDK was also used to monitor CPU and memory demand of the server side in the concurrence tests.

In the first test scenario, requests with different network types (Wi-Fi, 3G, 2G and 2G under low coverage) were tested. These tests involved making rounds of 100 requests repeated under diverse situations (different locations and times). In Fig. 1 minimum, maximum and median request times of each network type are shown. As we can observe, response times when requests are made via Wi-Fi and 3G connections are very satisfactory. In good coverage 2G networks response times still remain noticeably restrained. In contrast, when coverage is low in 2G networks ("Bad 2G" in the graph), response times moderately increase. Even so, this times remain in the satisfactory and acceptable user experience ranges.

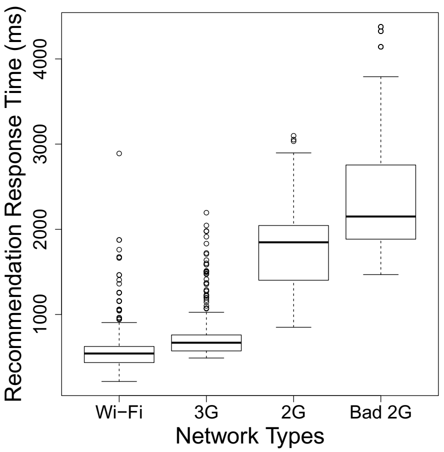

Fig. 1. Request times with different network types

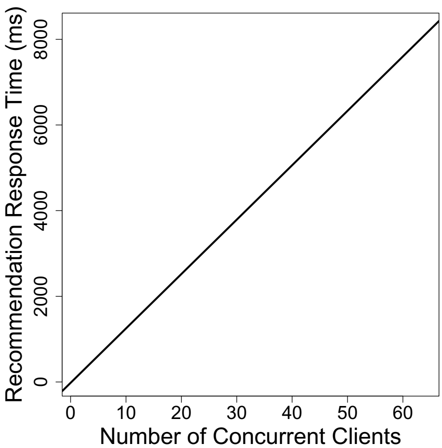

Fig. 2. Request times with concurrent clients

In the second test case, requests to the server with an increasing number of concurrent clients were made, up to 64 concurrent clients. Results obtained (see Fig. 2) show that even when around 35 concurrent clients are querying for recommendations, response time is acceptable. And attending about up to 60 concurrent clients could still be tolerable with a hardware equivalent to the one used in the tests if it is a punctual situation. Regarding CPU usage in this test cases, the lower concurrency scenarios show a linear increase, from around 10% in the single client test up to 30% in the 4 clients one. For higher concurrency cases, CPU demand increased at a slower pace, being the medium demand around 50% in the 64 concurrent clients test. Finally, memory consumption is quite constrained throughout all the test case even in the scenarios with a higher number of concurrent clients. In fact, memory usage remained similar through all the concurrence tests, with a maximum demand of around 250 MiB.

The third test case consisted in studying recommendation task response times with an increasing number of users registered in the platform. This aspect should have low impact in this response time, as the server only executes the rule engine against the data of the requested user. Therefore, response time increases as a function of the number of registered users only due to the query to the triplestore to retrieve this information being slightly slower. Server answering times with different numbers of registered users can be checked in Fig. 3.

In the last test scenario we measured the impact that the user having a different number of preferences defined has in the server response time. Preferences are the main element that rules use for grading communication means, so inference task will take longer for users who have a higher number of them defined. In Fig. 4 response times for users with different number of preferences are shown. As it would be expected, response times increase together with an increase in the number of preferences, though even for 50 preferences registered, which is an extreme case, this task takes around 2 seconds.

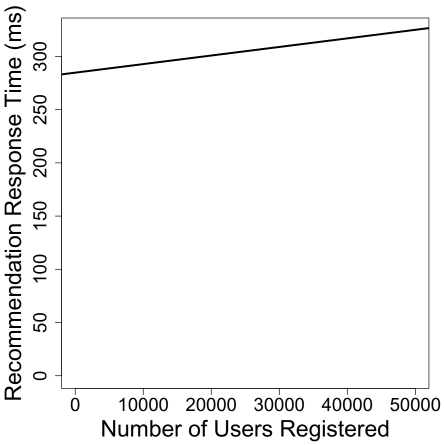

Fig. 3. Request times related to registered users number

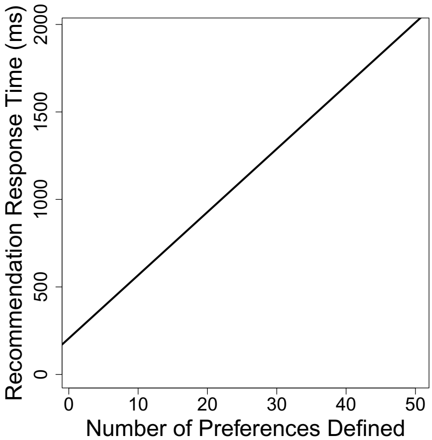

Fig. 4. Request times related to defined preferences number

5 Conclusion

In this article we have presented a platform that assists mobile users in the task of deciding which communication means to use to communicate with their contacts. This platform enables users to register their communication means and define different use preferences over them. Combining this preferences with users' context, the system is capable of inferring which communication means they would prefer to be contacted with in each situation.

Due to the real-time constraints of this system, we have also presented a series of tests aimed at assessing system performance executing this recommendation task. These tests have shown that the platform offers satisfactory response times even under poor mobile connections, with several communication preferences defined by each user and with both a considerable number of users registered in the system and concurrently working against it.

Having tested the technical viability of the proposal, as future work we plan to perform a large scale experiment with real users so as to evaluate the efficacy of the server, that is, if the communication means recommended by the platform in each situation are the most appropriate ones to get in touch with other users.

References

1. Abel, F., Henze, N., Herder, E., Houben, G.J., Krause, D., Leonardi, E.: Building blocks for user modeling with data from the social web. In: Proceedings of the International Workshop on Architectures and Building Blocks of Web-Based User-Adaptive Systems (WABBWUAS 2010), vol. 609, CEUR-WS. org (2010)
2. Apache Bench: ab - Apache HTTP server benchmarking tool, http://httpd.apache.org/docs/2.2/programs/ab.html
3. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Scientific American 284(5), 28–37 (2001)
4. Chen, R.C., Huang, Y.H., Bau, C.T., Chen, S.M.: A recommendation system based on domain ontology and swrl for anti-diabetic drugs selection. Expert Systems with Applications 39(4), 3995–4006 (2012)
5. Berrueta, D., Brickley, D., et al.: SIOC core ontology specification, http://rdfs.org/sioc/spec/
6. Brickley, D., Miller, L.: FOAF vocabulary specification 0.98, http://xmlns.com/foaf/spec/
7. Gao, Q., Abel, F., Houben, G.J.: Genius: generic user modeling library for the social semantic web. In: Pan, J.Z., Chen, H., Kim, H.-G., Li, J., Wu, Z., Horrocks, I., Mizoguchi, R., Wu, Z. (eds.) JIST 2011. LNCS, vol. 7185, pp. 160–175. Springer, Heidelberg (2012)
8. JConsole: Using JConsole: Java SE monitoring and management guide, http://docs.oracle.com/javase/6/docs/technotes/guides/management/jconsole.html
9. Knittel, J., Sahami Shirazi, A., Henze, N., Schmidt, A.: Utilizing contextual information for mobile communication. In: CHI 2013 Extended Abstracts on Human Factors in Computing Systems, pp. 1371–1376. ACM (2013)
10. Orlandi, F., Passant, A.: Enabling cross-wikis integration by extending the sioc ontology. LSSM+ (Cit. on p.) (2009)
11. Shaikh, J., Fiedler, M., Collange, D.: Quality of experience from user and network perspectives. Annals of Telecommunications-Annales Des télé Communications 65(1-2), 47–57 (2010)
12. Tang, J.C., Yankelovich, N., Begole, J., Van Kleek, M., Li, F., Bhalodia, J.: Connexus to awarenex: extending awareness to mobile users. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 221–228. ACM (2001)
13. Tao, K., Abel, F., Gao, Q., Houben, G.J.: Tums: twitter-based user modeling service. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) ESWC 2011. LNCS, vol. 7117, pp. 269–283. Springer, Heidelberg (2012)
14. W3C: OWL-Time: time ontology in owl, http://www.w3.org/TR/owl-time/
15. W3C: WGS84 Geo Positioning: an RDF vocabulary, http://www.w3.org/2003/01/geo/wgs84_pos#
16. W3C OWL Working Group: OWL 2 Web Ontology Language document overview, 2nd edn., http://www.w3.org/TR/owl2-overview/