Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 985–989
ISBN 978-83-60810-22-4

Cloud-based Assistive Technology Services

Ane Murua
Vicomtech-ik4
Mikeletegi Pasealekua, 57
20009 Donostia-San Sebastián,
Spain
Email: amurua@vicomtech.org

Igor González
Iriscom Sistemas S.L.
Torreatze 1D
20110 Pasaia San Pedro, Spain
Email: igonzalez@iriscom.org

Elena Gómez-Martínez
R&D Department,
Technosite-ONCE Foundation
C/Albasanz, 16
28037 Madrid, Spain
Email: megomez@technosite.es

Abstract—**Cloud computing will play a large part in the ICT domain over the next 10 years or more. Many long-term aspects are still in an experimental stage, where the long-term impact on provisioning and usage is still unknown.**

While first attempts at this field focused on service provisioning for enterprises, cloud is reaching individuals nowadays. Our proposal is to go a step further and, based on the proven benefits of the Cloud, improve Internet and technology access for those people always left behind when any technological progress takes place.

This paper presents the Cloud-based Assistive Technology Service delivering to individuals who face technology accessibility barriers due to ageing or disabilities. An example of how an Assistive Service is delivered to an individual in an easy and seamless way is given as a demonstration of how the future should be. This proof of concept has been developed within the INREDIS research project.

I. Introduction

CLOUD computing enables companies, public administrations and individuals, using networks such as the internet, to access their data and software on computers located somewhere else. It can help businesses – especially SMEs – to drastically reduce information technology costs, help governments supply services at a lower cost and save energy by making more efficient use of hardware. Cloud computing is already used widely, for example for web-based e-mail services.

Cloud computing has the potential to develop into a major new service industry, presenting great opportunities for European telecoms and technology companies. Client companies and public administrations can benefit from lower costs and state-of-the-art services by using cloud computing rather than installing and maintaining software and computing equipment of their own [1].

On the other hand, as our countries build out their broadband infrastructures to ensure that broadband reaches everyone, it is important that 'everyone' includes people with disability, literacy and aging related barriers to Internet use. We need to be sure that we don't stop at just connecting people to the Internet - but that we also see to it that they can actually use it, and benefit from all that it has to offer [2].

In other words, Cloud has much to contribute to Assistive Technologies' (ATs) ecosystem –Section II–. This paper presents part of the work done within the INREDIS research project –Section III–, namely, a basic architecture to be used for AT selection and delivering –Section IV– and a concrete use case developed for showing what would be the end result for users of the Cloud-based Assistive Technology Services of the future –Section V–. Finally, some future work is proposed –Section VI–.

The research described in this paper arises from a Spanish research project called INREDIS (INterfaces for RElations between Environment and people with DISabilities) [4]. INREDIS is led by Technosite and funded by CDTI (Industrial Technology Development Centre), under the CENIT (National Strategic Technical Research Consortia) Programme, in the framework of the Spanish government's INGENIO 2010 initiative. The opinions expressed in this paper are those of the authors and are not necessarily those of the INREDIS project's partners or of the CDTI.

II. The Assistive Technologies' Ecosystem

A. *What Does "Assistive Technology" Stand For*

According to the definition provided in ISO 9999:2007 "Assistive products for persons with disability -- Classification and terminology" [3], Assistive Products are understood to be any product (including devices, equipment, instruments, technology and software) specially produced or generally available, for preventing, compensating for, monitoring, relieving or neutralizing impairments, activity limitations and participation restrictions. Assistive Technology is technology used by individuals with disabilities in order to perform functions that might otherwise be difficult or impossible. Assistive technology can include mobility devices such as walkers and wheelchairs, as well as hardware, software, and peripherals that assist people with disabilities in accessing computers or other information technologies.

Moreover, according to the definition provided by the Class 22 of ISO 9999:2007 "Assistive products for communication and information", AT ICT products are understood to be devices for helping a person to receive, send, produce and/or process information in different forms. Included are, e.g., devices for seeing, hearing, reading, writing, telephoning, signalling and alarming, and information technology.

B. *The Assistive Technology Market*

The number of AT ICT products cataloged by European databases is well over 20.000, and the AT ICT industry in the EU is not a simple one. It is complex in various aspects, for example for the large number of products, for the large number of small firms, and for the different service provider systems that are used to get AT ICT products to disabled end-users.

However, one area common to the vast majority of firms is the marketing challenge: how to get the right product, via the right person, and with the right instructions and training to the disabled end-user who needs it. To some extent, this is a distribution and marketing challenge common to any industry, but in the AT ICT industry in Europe, the complexity of the different service provider systems is an extremely potent force in the marketplace [4].

First AT ICT products were dependent both from the platform and the device where they were installed. The only devices where ATs were used were computers, and the way for getting them was to purchase the AT into a local reseller, to receive the AT via mail or (the few times that it was possible) to download its installation program from the web.

This situation is slowly changing, but there are still large barriers for customers [5]:

Awareness

- End users are largely unaware of the available AT solutions.
- There is a lack of dedicated training in AT products and their capabilities, resulting in end users having AT they cannot use to the full extent, or in some cases not at all.
- ATs that are easiest to obtain are also the ones most abandoned.

Price

- High purchasing costs for end users are reported as a major barrier for wider deployment by disability organizations.

Mismatch between end user needs and offered AT

- End users are not provided with the required AT, resulting in a considerable percentage of obtained ATs being discarded within a year.
- AT that is being offered does not always satisfy the actual needs of the people with disabilities, hence their refusal to use them.
- According to some surveys, almost half of the end users experience problems using ATs.

III. INREDIS Basic Research Project

The work presented here belongs from the INREDIS project [6]. This project has carried out basic research in the field of accessible and interoperable technologies, with the scope of developing basic technologies that will enable the creation of communication and interaction channels between people with special needs and their environment.

A multidisciplinary team composed by 14 companies and 18 research partners has been involved in INREDIS The project began on 2007 and ended in December 2010.

INREDIS project aims a total integration of functional disabled users into the Information Society, including popular fields such as: domotic, urban and local mobility, shopping information, banks, digital television and other areas of interest.

The base of such approach is that a person with an adapted or personalized controller (where *controller* does mean a fitting of a proper User Interface into a suitable hardware) should be able to interact and control different services and devices by means of an interoperability architecture. Using a personalized controller, the achievement of a proper control is eased significantly as the accessibility problem of the whole environment is reduced to just solving, if any, the accessibility issues with the user controller.

A prototype can be seen in [7] showing the above (see Fig. 1).

In this example (further described in [8]), a pluggable User Interface for people with visual impairments can be seen, which is based on a DHTML page running on a vertically handled tablet-PC .

This DHTML page is rendered in a web browser. Popular browsers, such as Microsoft Internet Explorer, Mozilla Firefox or Safari, are supported. The DHTML code has been correctly tagged so that it is compatible with screen readers such as Jaws [9]. An Universal Control Hub (UCH) [10] acts as a gateway allowing pluggable UIs to remotely control the TV set.

A. Proposed Architecture

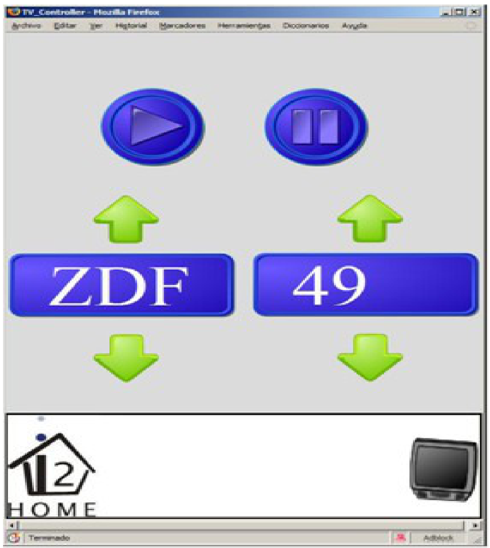

Fig 1. DHTML based UI for accessible TV control

Fig. 2 (next page) shows the INREDIS architecture at high-level.

Firstly, user's device communicates to INREDIS architecture core, which connects up *Interoperability Platform*, containing different interoperability protocols (i.e. UCH, OSGi and Web Services).

The Interoperability Platform selects the corresponding protocol in order to interact with a specific device or service.

Once the service is known, it is necessary to adapt its user interface. So, the user’s device information and the user’s profile are asked to the *Adaptive Modeling Server*, which looks them up into the *Knowledge Base*.

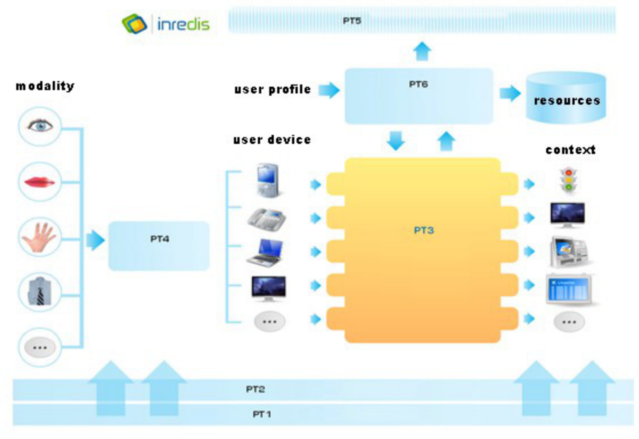

Fig 2. INREDIS Architecture (High-level)

This information, together with the context information, is headed for the *AT Middleware*, which queries to *Knowledge Base* the most appropriate AT and delivers it as a service (SaaS approach) [11].

All collected information is passed to Interface Generator in order to provide the most suitable user interface.

IV. INREDIS AT MIDDLEWARE

The AT Middleware is the module responsible for providing AT services within the INREDIS architecture.

Unlike the legacy purchase methods introduced at Section II, the AT Middleware instantly provides the user the right ICT AT with his desired configuration.

In this sense, as seen in Fig.3, if an AT delivery service does exist within an architecture that aims to simplify the interaction of people with disabilities with the devices and services of their environment, why not export this service and use it to solve disability, literacy and aging related barriers to Internet use as exposed on Section I?

Fig. 4 shows AT Middleware's sequence diagram when finding an AT within INREDIS.

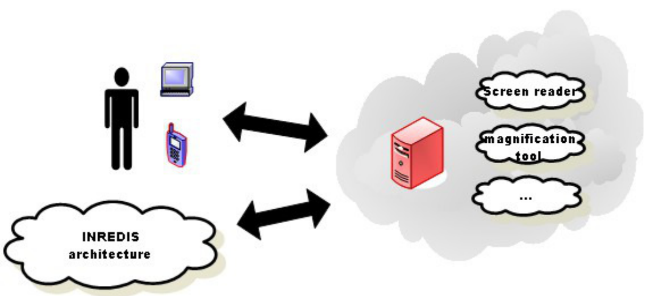

Fig 3. Using the AT Middleware (right) directly (up) and via INREDIS architecture (down)

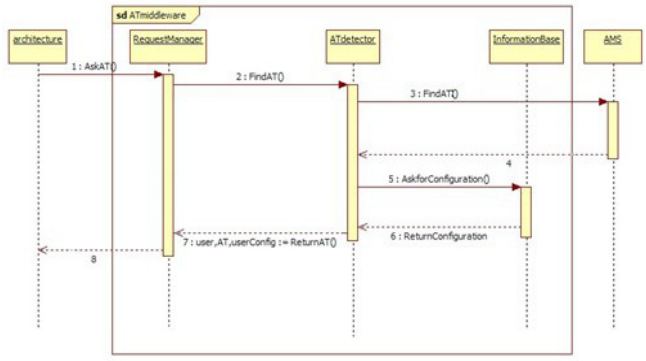

Fig 4. Using the AT Middleware (right) directly (up) and via INREDIS architecture (down)

As seen on the diagram, the AT Middleware is composed by 3 modules: Request Manager; AT Detector; AT Information Base.

The Request Manager is an OSGi bundle. It exposes a Web Service, which is the main entry point of the AT Middleware.

The AT Detector OSGi bundle queries the AT ontology [11] in order to find the most appropriate AT for an user, taking in mind its needs and preferences.

And last, the AT Information Base is composed by a database, where users' personal AT configurations are stored, and an OSGi bundle, which queries the database when needed.

But this approach won't be complete until a wide collection of ICT ATs in the cloud appear in the market. Once this is achieved, AT users will gain the following advantages:

- Access to a wider range of products (multiple products of the same type / multiple product types)
- Access to free AT ICT products
- Access to low-cost products, thanks to the pay-per-use approach
- No need to install a software for every AT product
- Seamless AT execution and use (no need for complex setup procedures or for the execution of various commands continuously)

At INREDIS project, in order to incorporate the first AT ICT products to the platform and to demonstrate that it is possible to give an AT service based on the cloud, we have chosen four common ATs, i.e. a multi-language translator, a text-to-speech (TTS) synthesizer, an avatar who communicates through sign language and an automatic captioning engine [12]. All of them have been successfully transferred to the cloud.

The AT used into the end-to-end implementation presented at Section V is the multi-language translator.

Based on the Google Translate [13] API [14] provided by Google, two cloud-based AT services have been developed. Both services are remotely accessible Web Services: one of them follows a REST approach while the other is the SOAP version of the service.

Those multi-language translation Web Services contain a single method called *translate*. Its input parameters are *textToTranslate*, *sourceLanguage* and *targetLanguage*, while the output is a string containing the translation done by the service.

V. Proof of Concept: Seamless Assistive Technology Delivering

As a proof of concept of cloud-based AT service delivering, an end-to-end implementation has been done. The goal is to show how the system would look in the eyes of the user, i.e., to demonstrate the benefits of this new market paradigm.

So, an Android application (app) has been adapted in order to make it capable to use cloud-based multi-language translation service when needed. The chosen app is the *Bluetooth Chat* app included into the Android SDK [15].

Basically, three main changes have been made into the original app: (1) communication with the INREDIS AT Middleware added; (2) localized resource sets for showing the UI into the language that the user has set its phone; and (3) the ability to use the cloud based multi-language translation cloud-based AT services developed. The end result for the user is that, once logged into the AT Middleware, if he wants to use the Bluetooth Chat in a place where he won't be understood, the app will be aware of that, switching on the translation without user intervention.

More on detail, the changes made into the Bluetooth Chat app have been the following:

- (1) Communication with the INREDIS AT Middleware:

This process should be as least invasive as possible for the user, even running in the background if possible.

This issue has been solved by developing another Android app (as Fig. 5 shows) that logs the user into the AT Middleware (see Fig. 6).

This app queries the user's profile in order to get its preferences and, when launching the Bluetooth Chat, sends this preferences to it. This way, the Bluetooth Chat knows whether it has to change its original operation mode and start using the cloud-based translation AT service or not.

- (2) Localized resource sets for showing the UI into the language that the user has set its phone:

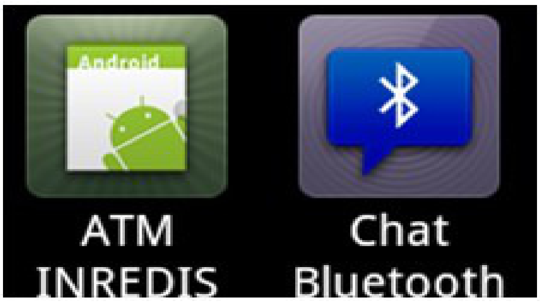

Fig 5. AT Middleware app (left) and Bluetooth Chat app (right)

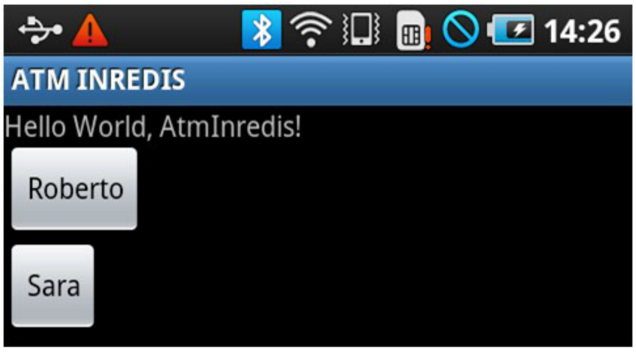

Fig 6. User login UI

The Bluetooth Chat app is, by default, in English. It makes no sense to give an application in English to a user who needs support to communicate in other languages than its own native language. For this reason, we have modified the original app, adding localized resources that make possible the use of the user's preferred language by the UI (e.g., menus, info messages,…) of the app.

We have assumed that user's preferred language is the one selected into the locale configuration of its mobile phone. By default, the Android system selects which resources to load, based on the device's locale [16]. So, multiple resource sets on different languages have been added to the original structure off the app, creating different *res/values-xx* folders, (where *xx* is the code of each different language).

If greater levels of customization want to be achieved, the language in which the user has configured its mobile phone can be bypassed. This way, the incoming information from the AT Middleware would be used to set the language of the Bluetooth Chat's UI.

- (3) The ability to use the cloud based multi-language translation cloud-based AT services developed:

Finally, it is necessary to integrate the translation service into the chat application. The intended result is that two people can communicate in their native or preferred languages. I.e.:

Being *A* an Spanish user, and *B* an English user of the system, both users run the Bluetooth Chat app in their phones and pair their devices via bluetooth. *A* writes on her device a message in Spanish and presses "enviar" (send) button. *B* receives *A*'s message in English. *B* writes on his device a message in English and presses the "send" button. *A* receives *B*'s message in Spanish.

In order to make this scenario possible, the multi-language translation Web Services previously developed have been seamlessly integrated. This way, when user clicks the "send" button, a petition is sent to the translation Web Service, containing the message to be translated and the languages of origin and destination. The petition is processed on the cloud instantly and the translated message is returned and delivered to the recipient.

Since the language translation Web Services developed can be invoked via SOAP or REST, the procedures for remotely accessing them from the chat application differ one from another.

Android's own libraries are used for querying the REST Web Service for a translation. In particular, the android.net.Uri [17] class is used, which eases the building of the URI of the call from the various parameters involved (e.g.: http://192.174.15.29:8080/GoogleTranslatorREST/ resources/methods?textToTranslate=hello&sourceLanguage=en& targetLanguage=es) and processes its response.

In the case of the SOAP Web Service, Android has no specific libraries to request the Web Service. The option chosen is the use of the library ksoap2-android [18].

Finally, Fig. 7 shows the final appearance of the Bluetooth Chat app when using the cloud-based multi-language translation Assistive Technology service:

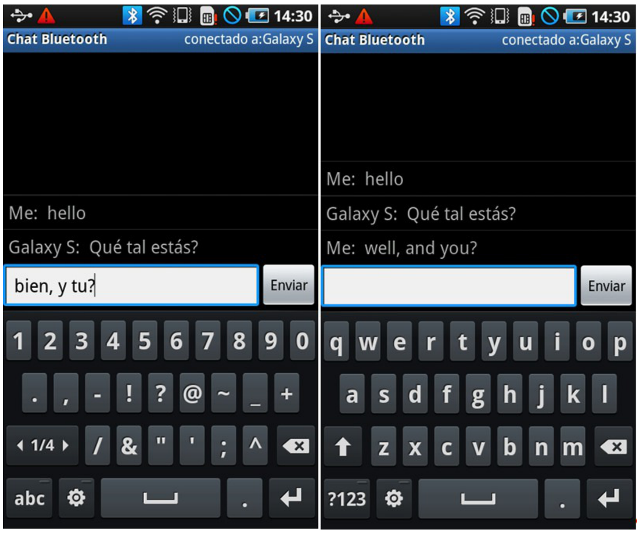

Fig 7. User sends chat message while Spanish to English translation is on

VI. Future Work

Once demonstrated that the Cloud-based Assistive Technology service delivering approach is feasible, it still remains a lot of work to do in order to see this system in the market in a short period of time.

The most important work to do is to progressively migrate all the ICT ATs to the cloud. In this sense, the resources that AT developers have available are limited, so, migrating and supporting the actual users in parallel will be a big challenge for them.

There are some efforts trying to standardize the accessibility APIs present on different operating systems. Once this achieved, the task of migrating ICT ATs to different OS and to the cloud will be much easier.

References

[1] N. Kroes, "Digital Agenda: Commission seeks views on how best to exploit cloud computing in Europe": http://europa.eu/rapid/pressReleasesAction.do?reference=IP/11/575&format=HTML&aged=0&language=EN&guiLanguage=en
[2] GPII project website: http://gpii.net/
[3] International Organization for Standardization. ISO 9999:2007 "Assistive products for persons with disability -- Classification and terminology"
[4] Stack, J., L. Zarate, C. Pastor, N.-E. Mathiassen, R. Barberà, H. Knops, H. Kornsten (2009) Analyzing and federating the European assistive technology ICT industry, Final Report, March 2009.
[5] Lopes, R., Bandeira, R., Carriço, L., Van Isacker, K. (2010). Towards Mobile Web Accessibility: Vision And Challenges. Proceedings of the first International ÆGIS Conference - 7-8 October 2010 (pp. 151-158).
[6] INREDIS project website: www.inredis.es
[7] URC Based Accessible TV video. http://youtu.be/9xGMTuVZRkY
[8] Epelde, G. et al.: URC based accessible TV. In: Proceedings of the 7th European conference on Interactive Television (EuroITV 2009), pp. 111–114. ACM, New York, NY, USA (2009)
[9] Freedom Scientific. Jaws for Windows website: http://www.freedomscientific.com/fs_products/software_jaws.asp
[10] Zimmermann, G., and Vanderheiden, G. 2007. The Universal Control Hub: An Open Platform for Remote User Interfaces in the Digital Home. In Proceedings of Human-Computer Interaction International 2007 (Beijing, P.R. China, July 22-27, 2007). HCI'07. Springer, Heidelberg, LNCS 4551, 1040— 1049.
[11] Iglesias-Pérez, A. et al. 2010. A Context-Aware Semantic Approach for the Effective Selection of an Assistive Software. In proceedings of the IV International Symposium of Ubiquitous Computing and Ambient Intelligence (UCAml 2010).
[12] INREDIS project. D5.5.1 deliverable
[13] Google Translate website: http://translate.google.com
[14] Google Translate API on Google code: http://code.google.com/apis/language/translate/overview.html
[15] Android Developers, "Bluetooth Chat sample code": http://developer.android.com/resources/samples/BluetoothChat/index.html
[16] Android Developers, "Localization": http://developer.android.com/guide/topics/resources/localization.html
[17] Android Developers, "Uri public abstract class": http://developer.android.com/reference/android/net/Uri.html
[18] ksoap2-android library: http://code.google.com/p/ksoap2-android/