

Virtual Assistant Platform

Damjan Kužnar, Aleš Tavčar and Jernej Zupančič
 Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
 E-mail: damjan.kuznar@ijs.si

Mihai Duguleana
 Transylvania University of Brasov Brasov, Romania

Keywords: virtual assistant, SaaS, cloud application

Received: June 15, 2016

In this paper a novel virtual assistant platform is presented. The platform allows for easy creation, administration, and use of virtual assistants as additional services on websites and as a stand-alone application on mobile platforms. The implementation is based on the principle of Software as a Service, which significantly eases the installation and maintenance of the system and consequently reduces costs.

Povzetek: Prispevek predstavlja platformo za virtualne asistente s poudarkom na spletni aplikaciji.

1 Introduction

Websites of major companies, various institutions, ministries and local administrations are often very large and have a very complex navigation structure, which makes it difficult for visitors to find the information they need. This problem is addressed by using the virtual assistant that helps the visitors find the information in natural language interaction. This kind of interaction hides the complex structure of the website in the background by providing the answers to the questions asked by the users. The use of virtual assistant allows visitors to search for information in a more natural way, which is especially important for less technically experienced users.

In this paper we describe a virtual assistant platform that has been developed within the project Assistant [1] and is currently running more than 200 virtual assistants. Modern guidelines for development and software distribution were taken into account during the development and, therefore, the platform was developed as a web service that runs in the cloud. The assistant represents the core for several ongoing projects dealing with tourists, municipalities and smart cities.

2 Virtual assistant platform

The service, which supports the creation, management and use of virtual assistants, has been developed on the principle of Software as a Service (SaaS) [2], which brings a number of benefits to customers. For example there is no need for the customer to purchase and maintain hardware, as software runs in the cloud, and no need for installing and maintaining the software. The system consists of the following components (Figure 1):

- Web service in the cloud, which allows subscribers and end-users ease of use through the SaaS principle.

- Web client that offers visitors of a website (end-users) a rich graphic interface for communication with a virtual assistant.
- Mobile clients in the form of mobile applications for the Android, iOS, BlackBerry and Windows Phone 8 platform, which enable for easy communication with a virtual assistant through mobile phones.
- Administrative tools for managing virtual assistants, which allow customers to adapt their assistant to their individual needs. Tools are accessible through browser and are password protected.

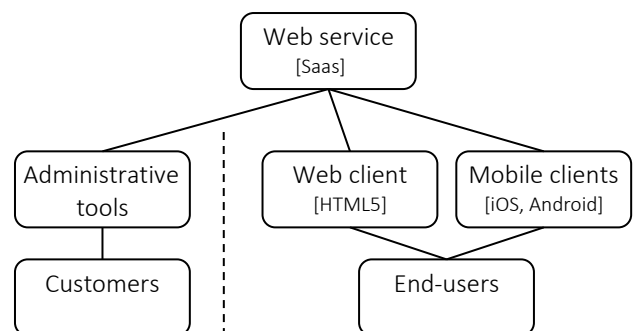


Figure 1: Virtual assistant platform overview.

3 Web service

Modularity, based on the principles of multi-agent systems [3], is an essential feature of the system. This is reflected in such a way that all system functions are implemented as independent modules (i.e. agents) and are able to communicate to each other. The system core,

which plays a similar role as the platform in multi-agent systems, provides the mechanism for communication between modules. The components of the system (Figure 2) will be described below.

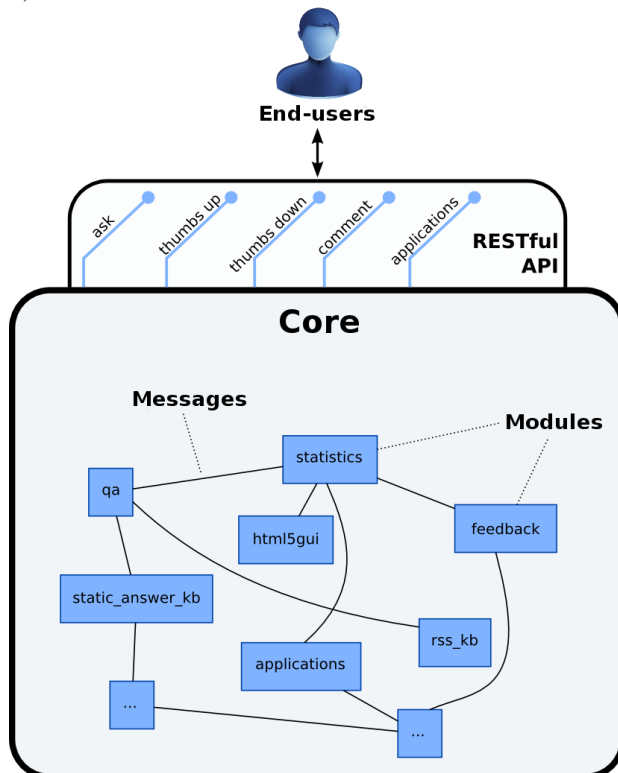


Figure 2: The modular design of the system.

3.1 Core

The system core implements the following functions, allowing the modules to extend the basic functionality of the system:

- Module management, including discovery, installation, updating, starting and stopping of the modules. The virtual assistant platform has a broad collection of modules available, which can be stopped if desired and thus speed up the system.
- Registration and triggering of events, and registration of listeners that receive triggered events. The modules that trigger the events also define an arbitrary message structure that facilitates communication between modules.
- RESTful API endpoints [4] definitions. Each module can define a function that can be used by end-users (through clients such as web client or mobile application).
- Relational database access. The modules can store the necessary data into a relation database.
- File system access. The modules that need to write and read files are provided with the required access rights.

3.2 Modules

The modules are independent functional components that extend the basic set of system capabilities. The core modularity allows the system to be extended to perform any task, however, this article describes the modules for the virtual assistant. Among the most important modules are:

- *qa*: a module that receives questions from the user, forwards it to the answering modules (*_kb modules) and chooses the most appropriate response on the basis of a predefined procedure.
- *_kb (eg. *static_answer_kb*): are modules for question answering that obtain answers in various ways. For instance *rss_kb* obtains answers from RSS feeds.
- *html5gui*: a module that implements a web based graphical interface of a virtual assistant that users can interact with on websites.
- *applications*: a module for managing applications (each module can define its own application) that virtual assistant provides to end-users in addition to the basic functionality of question answering.

The existing system also hosts a number of other modules that provide answers to questions, facilitate the administration of the system, record the conversations between users and virtual assistant, etc.

3.3 Messages

Messages between the modules are transmitted when events occur within the system (e.g. when a question is received). Events can be registered and defined by the system core or by the modules that also define the structure of messages that will be sent when the event is triggered. The modules that have registered to that certain event receive the message. Those modules can then return the results of their processing back to the module that triggered the event. This enables the creation of complex processing procedures.

3.4 REST API

It is essential for the system to be able to communicate with the outside world. The REST specification [5] is used to create the API in the form of web services. All communication between users and the system goes through this interface, usually by the means of various graphical clients such as HTML5 client for websites and apps for mobile platforms.

The most important functions provided by the API for virtual assistants are:

- /ask (HTTP GET method)
 - Input parameters:
 - question: a question that we want the assistants to answer.
 - context (optional): the context within which the question was stated (eg. in applications and

forms of municipal administration). Available contexts are obtained through the function / applications.

- Response:
 - answer: answer of a virtual assistant in HTML format.
 - id: serial number of the answer, used for storing the evaluation and comments on the answer.
 - URL (optional): a website that is associated with the response. In the case of using a web client, the page will automatically open in the background, in the case of mobile applications the user can open a link in the integrated browser.
- / vote / id / Up (HTTP POST method)
 - Input parameters:
 - id part of the route replaced with the value we received as part of the response of the function call / ask. It represents the number of the answer. Example: a call to / vote / 42 / up corresponds to the positive assessment of the response with the serial number 42.
 - This end point has no response
- / vote / id / down (HTTP POST method)
 - similar to / vote / id / up, only that in this case a negative rating is recorded. In the case of multiple ratings for the same answer the most recent rating is stored.
- / comment / id (HTTP POST method)
 - Input parameters:
 - id: number of the answer, which refers to a comment.
 - comment: comment on the answer.
 - name_and_surname: name and surname of the commentator.
 - email (optional): E-mail of the commentator.
 - This end point has no response

All functions return the answers in JSON notation [6], unless the client defines the callback input parameter, in which case the answer is returned in JSONP notation [7].

4 Web client

Web client is used to integrate the virtual assistant in the customer's website. The assistant can appear

automatically when the user visits a website or when the users clicks on a link that starts the assistant.

On launch the web client (Figure 3) hovers over the content of the website and can be freely moved, allowing the user to continue browsing the web page in the background. Upon entering a question into the text box, the web client calls the web service to obtain an answer, displays it in the box below the input field and, if provided by the administrator, opens a webpage in the background that is related to the answer. In this way the web client displays a summary of the response and the visitor is invited to obtain more detailed information from the website in the background if he or she is interested. The Slovenian text in Figure 3: Who is the mayor? The major of Pivka municipality is Mr. Robert Smrdelj ...



Figure 3: Example of a virtual assistant for the Pivka municipality.

In addition to the common virtual assistant functionality our platform also introduces the concept of applications or apps (accessible via the menu). Apps extend the basic functionality by enabling the use of specific services, e.g. ticket booking, or by focusing the search for answers on specific areas, e.g. the municipality contact information. Every customer can configure its own set of applications that it wants to offer to the visitors.

5 Mobile apps

Communication with the virtual assistant is also possible via mobile devices. The interface was adjusted for small screens, since the use of the web interface on mobile

devices is awkward, especially for technically unskilled users. The mobile applications are important for several reasons. Mobility is the most important reason, since the user can use the assistant from anywhere, for example the user can ask “Where can I find a pharmacy?” when on the move in a municipality. In this case the assistant would find the information and offer to the user a list of pharmacies in a given municipality with the associated opening time.

We have developed mobile applications for the two most popular mobile platforms: Android and iOS. Both mobile applications have the equivalent functionality to the web client graphical interface. Figure 4 shows the mobile applications screen design for Android platform. On the upper side of the screen is an input box where the user enters a question. Below the input field is the area for the answer, and underneath is the toolbar, where the user can find all the additional functionalities. The Slovenian text in Figure 4 corresponds to: Municipality Pivka; Do you have any question for me?; Ask me anything regarding municipality such as

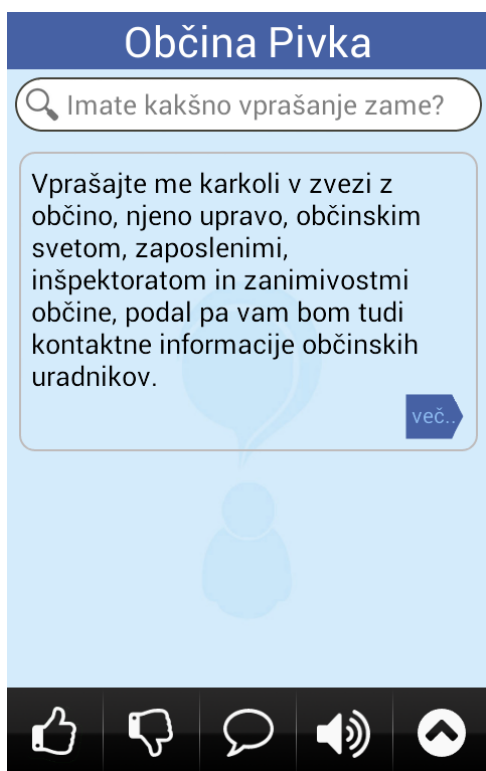


Figure 4: The main screenshot of the Android app.

The functionality of mobile application is identical to the one in the web interface. The user enters the question in the search box and the application sends a query to the server that returns the most appropriate response. In the bottom toolbar there are additional assistant’s features such as: voting on a particular response (positive or negative); posting a comment on a particular answer; enabling speech synthesis; and allowing the users to select one of the applications offered by the assistant.

Due to smaller screen sizes and data transfer restrictions on mobile devices some specific functionality is changed and adapted for mobile devices. The website

that which contains more detailed information is accessible by clicking the “more...” button (in contrast to the web client that displays the website in the background). The app also contains the list of all available assistants and the user can select which assistant it wants to connect to. The list of assistants appears when the user opens the app for the first time but is also accessible via the app settings. Further mobile applications are adapted for the elderly and visually impaired, because in addition to speech synthesis it is also possible to increase the font size of the entire graphic interface.

With the development of mobile applications we wanted to bring our virtual assistant platform even closer to users and to allow its use on a daily basis.

6 Conclusion

In this paper we presented a virtual assistant platform – framework for the creation and maintenance of virtual assistants.

The service operates on the advanced web technology principles. The modular design enables to easily extend the service and add new functionality. Therefore, this service is suitable for a wide range of customers from municipalities and associations to public and private companies, which have their own specific requirements.

7 Future work

Due to its extensibility the developed virtual assistant platform can be used in various domains. We will exploit the modular architecture in several upcoming projects. Some of the existing modules will be reused, however, additional modules, specific to each project, will also be developed. In the following paragraphs the important projects will be described and their contribution to the virtual assistant platform will be presented.

7.1 AS-IT-IC

Austrian-Slovenian intelligent tourist information center (AS-IT-IC) project was accepted in the cross-border Cooperation Programme Interreg V-A Slovenia-Austria in the programme period 2014-2020. The project addresses the problem of not getting the desired information about natural and cultural heritage sites in the Slovenian-Austrian cross-border area in an integrated way. The goal of the project is to create a joint Austrian-Slovenian center - an ICT supported network of service providers and tourist offices, municipalities, tourists and citizens to enhance continuous cooperation between them. Virtual assistant will be one of the ICT tools that will support the AS-IT-IC center. It will provide automatic answering in natural language to the questions related to the heritage sites and various other relevant tourist information, therefore, relieving the tourist information officers from providing answers to repetitive questions. The system will, in cooperation with human operators (tourist information officers), provide all the relevant tourist information and help the tourists plan a

multiday trip in the Slovenian-Austrian cross-border area.

During the project several new modules will be added to the platform: path planning interface; intelligent search over the content on natural and cultural heritage; automatic information extraction of knowledge from existing websites with relevant content, etc.

7.2 eHeritage

The goal of the project is the preservation and restoration of nation's cultural heritage in electronic form. With the advancements in the field of virtual reality, intelligent systems, and AI methods, experts can now use augmented and virtual reality, 3D immersive graphics and intelligent GUIs to restore and reproduce historical sites. Interested individual can experience the historical sites in novel interactive and informative ways.

Virtual assistant platform will provide the means for information retrieval from existing heritage sources in a natural language. During the project the following new modules will be added to the platform: natural language museum navigation; automatic extraction and intelligent content analysis of existing heritage databases, etc.

8 Acknowledgement

The work was co-funded by: the Slovenian Ministry of Education, Science and Sports through a public call for funding the development of e-services and mobile applications for public and private non-profit organizations 2012-2013 (JR ESMA JZNO 2012-2013); Cooperation Programme Interreg V-A Slovenia-Austria 2014-2020, project AS-IT-IC; European Union's Horizon 2020 research and innovation programme under grant agreement No. 92103, project eHERITAGE (Expanding the Research and Innovation Capacity in Cultural Heritage Virtual Reality Applications).

9 References

- [1] "Projekt Asistent." Domov - Projekt Asistent. N.p., n.d. Web. 23 Nov. 2016. <http://www.projekt-asistent.si/>
- [2] Choudhary, Vidyanand. "Software as a service: Implications for investment in software development." System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on. IEEE, 2007.
- [3] Ferber, Jacques. Multi-agent systems: an introduction to distributed artificial intelligence. Vol. 1. Reading: Addison-Wesley, 1999.
- [4] Richardson, Leonard, and Sam Ruby. RESTful web services. " O'Reilly Media, Inc.", 2008.
- [5] Fielding, Roy Thomas. Architectural styles and the design of network-based software architectures. Diss. University of California, Irvine, 2000.
- [6] "Introducing JSON." JSON. N.p., n.d. Web. 23 Nov. 2016. <http://www.json.org/>
- [7] "Safer cross-domain Ajax with JSON-P/JSONP". JSON-P.org. N.p., n.d. Web. 30 Oct. 2011. <http://www.json-p.org/>

