Sequence analysis

**mORCA: sailing bioinformatics world with mobile devices**

Sergio Díaz-del-Pino¹, Juan Falgueras², Esteban Perez-Wohlfeil¹ and Oswaldo Trelles¹,*

¹Department of Computer Architecture and ²Department of Computer Languages, University of Malaga, 29071 Malaga, Spain
*To whom correspondence should be addressed.

Abstract

**Motivation:** Nearly 10 years have passed since the first mobile apps appeared. Given the fact that bioinformatics is a web-based world and that mobile devices are endowed with web-browsers, it seemed natural that bioinformatics would transit from personal computers to mobile devices but nothing could be further from the truth. The transition demands new paradigms, designs and novel implementations.

**Results:** Throughout an in-depth analysis of requirements of existing bioinformatics applications we designed and deployed an easy-to-use web-based lightweight mobile client. Such client is able to browse, select, compose automatically interface parameters, invoke services and monitor the execution of Web Services using the service’s metadata stored in catalogs or repositories.

**Availability and implementation:** mORCA is available at http://bitlab-es.com/morca/app as a web-app. It is also available in the App store by Apple and Play Store by Google. The software will be available for at least 2 years.

**Contact:** ortrelles@uma.es

**Supplementary information:** Source code, final web-app, training material and documentation is available at http://bitlab-es.com/morca

1 Introduction

The popularity of mobile devices, such as smartphones and tablets, have broken the barrier to ubiquitous access to bioinformatics and biomedicine (BIBM) web services. However, intrinsic device limitations such as the screen size, the touch paradigm or the distractions to which medical staff are subjected (Deegan, 2013) radically change with respect to traditional WIMP (Windows, Icons, Menus and Pointer) interfaces (Harrison *et al.*, 2013) in desktop computers. Therefore, devices demand not only the adaptation of the software clients to access services, but also the rethinking of their functionality and characteristics. Furthermore, the variety of the devices available usually makes the development dependant of the platform. In this sense, we have developed mORCA as an easy-to-use, hybrid web-based, platform independent and lightweight mobile client to allow the access, composition, invocation and monitoring of proprietary and third party web services, such as those offered by EBI and NCBI.

2 Materials and methods

mORCA has been designed with a modular architecture based on a multi-layer scheme in order to make it easy to extend it both with new repositories and new services. Two main components can be easily distinguished from this design: on one side the module that supports MAPI (Karlsson *et al.*, 2013) a system that standardizes web services access through a common interface and also providing an upper layer that allows access to web services through simple HTTP queries. On the other component, we find the set of modules that compose the client itself. Such modules accomplish already mentioned tasks: authentication, browsing, intelligent services search, parameter dialog composition, enacting, monitorisation and file managing.

Our client is using web technologies such as jQuery (jQuery Project, 2010a), VanillaJS, and jQuery Mobile (jQuery Project, 2010b) as frameworks to optimize the mobile device experience. NodeJS (Dahl, 2009) and MongoDB (MongoDB, 2009) are used as
middleware to monitoring services, while the access layer to connect with MAPI is written in Java. SOAP is the main protocol for most of the client-server transactions.

mORCA incorporates the following main characteristics: (i) Being multi-platform (It has been validated in iOS, Android, and is able to run in any device which includes a browser); (ii) Centralized authentication using the MAPI WS; (iii) Browsing and service discovering through different repositories using an intelligent search engine; (iv) Responsive interfaces for each service dynamically generated from service’s metadata stored in the repository; (v) Execution and monitoring of services and (vi) Modular and expandable design.

As a simple example of performance, we present a typical bioinformatics use case that starts with the execution of a service to get a Uniprot sequence, and after saving it in our cloud-based file system, a BLAST service is run to find similar sequences (this exercise is step-by-step defined in the Supplementary material). The BLAST service is registered at EBI and demonstrates how third party service can be also invoked. In the Figure 1A, we can observe a general overview of the app. After browsing through the repository, we can get (Fig. 1B) the list of available services and once BLAST is selected, the app retrieves the metadata corresponding to this service, including both the parameter and data types, and depending on each type of data, an automatic adaptation to the screen size is performed (Fig. 1C) to finally compose a complete input parameters dialog. Once the program is executed we can (Fig. 1D) trace its execution live and preview its results.

3 Conclusion

It is not difficult to perceive that mobile devices are here to stay. Current trends include their incorporation to many tasks that where limited to personal computers, and for which mobile devices can contribute with their intrinsic characteristics: ubiquitous access, online access to current information, cooperative activities, etc. Moreover, BBMs should take advantage of the depicted scenario by incorporating standards into their applications and hence improving interoperability. In this document we have presented a mobile app that is able to easily enlarge the portfolio of services to execute through a well-defined standard based on repositories. Furthermore, this method enables the app to explore repositories, search services, dynamically compose responsive interfaces, execute and monitor tasks and visualize results. Its modular design enables expansion towards new repositories, data types and visualization methods. Moreover, it is platform independent, and can thus be used on any mobile device.

These properties make mORCA able to provide both regular users and system administrators with the possibility of using and providing services with different data types and formats under the same application, with a common interface to avoid the tremendous efforts required in porting and maintaining these individually.

Acknowledgements

The authors of the manuscript wish to thank the BITLAB Group and the application testers for all their support and comments which have significantly contributed to improve this work.

Funding

This work was partially supported by the Instituto de Salud Carlos III [PT13.001.012, RD12.013.006], the ELIXIR-EXCELERATE Project [INFRADEV-1-H2020 676559] and the University of Malaga.

Conflict of Interest: none declared.

References

Dahl,R.L. (2009) Nodejs (20 July 2017, date last accessed).
Deegan,R. (2013) Managing distractions in complex settings. In: Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services, MobileHCI ’13, ACM, New York, NY, USA. pp. 147–150.
Harrison,R. et al. (2013) Usability of mobile applications: literature review and rationale for a new usability model. J. Interact. Sci., 1, 1.
jQuery Project, T. (2010a) jQuery (20 July 2017, date last accessed).
jQuery Project, T. (2010b) jQuery mobile (20 July 2017, date last accessed).
Karlsson,J. et al. (2013) MAPI: a software framework for distributed biomedical applications. J. Biomed. Semantics, 4, 4.
MongoDB (2009) Mongoddb (20 July 2017, date last accessed).