Ubiquitous Access to Deep Content via Web Services

Zsolt Tivadar Kardkovács[1] and Gábor Mihály Surányi[2]

[1] Budapest University of Technology and Economics,
Department of Telecommunications and Telematics,
H-1117, Budapest, Magyar tudósok körútja 2., Hungary
[2] University of Karlsruhe,
Institute for Program Structures and Data Organization,
Am Fasanengarten 5, D-76128 Karlsruhe, Germany

Abstract. Current research activities in the computer science area focus on integration of systems with diverse purposes and architecture. They include the usage of mobile devices just like any immobile computer and the efficient, combined presentation and utilisation of contents from multiple providers together on the web. Even though these two tasks are naturally tackled in different ways, in this paper we propose a single solution to both problems based on logic, which, furthermore, makes their seamless integration possible. That is, without scrapping the well known models that have proven good, we offer a scalable architecture that supports wireless access to the most valuable pieces of the Internet keeping parts of the existing infrastructure and with no duplication of functionality.

1 Introduction

They provide long-distance and in-motion communications, however, mobile phones have raised novel problems for the technical experts since there is a natural demand to be able to make use of the everyday services anywhere and at any time. At the same time the Internet and its main application, the World Wide Web gained increasing popularity and support as an inexhaustible knowledge store.

Nowadays the technology of connecting a mobile device to the Internet is fully elaborated. The mobile device is always linked to a base station. It is the proxy of the mobile service provider, that requests web pages on behalf of the customer and forwards them to the base station. Data can be sent to the cell phone via two different software protocols. One of them is the native language of the Web, the HTTP. The other one is the WSP, which has been specially developed for mobile devices. Although theoretically the original HTTP content is delivered to the mobile devices, in practice it is not. The main reason is that most of the mobile devices have limited capabilities compared to a usual computer (eg. they have smaller display, they do not run Java applets). After all, it is noticeable that web servers present the same content in two different formats for mobile and

J.M. Cueva Lovelle et al. (Eds.): ICWE 2003, LNCS 2722, pp. 208–211, 2003.

immobile clients. It results in unnecessary duplication, which makes the system less manageable. To solve this problem, gateways are built between protocols and languages. It can not be a permanent solution, however, because those protocols and languages are evolving as well. In the near future we can not even expect that mobile devices become capable of displaying most web pages for the protocols adaptable to bandwidth and display environment are still missing.

The increase of data and information surrounding us has changed the operation of the Web qualitatively. There have been content providing databases installed behind the web servers [1]. Databases only manage the data, which are formatted by the web servers on the fly in response to client queries. This architecture is not only unsuitable for mobile access because of the restricted capabilities we mentioned before, but it is also difficult to integrate the knowledge of systems about different subjects due to the human-oriented, look-centric interface.

According to researches there are two ways to (at least) virtually consolidate the data sources on the web. One of them is more technical and invents web services, which have proven good in proprietary systems. The protocols and languages of web services (XML, WSDL, SOAP, UDDI etc.) are standardised but have different views of modelling so each service is unusable beyond its direct purpose. The other idea attaches further data to all data on the web so that computers can also understand them, that is the Semantic Web. A convergence between the two approaches is emerging [2], we are going to amplify it.

An unprecedented mathematical formalism F-logic[3] was published in 1995. Its creators' aim was to unify the object-oriented paradigm, the so-called frames of artificial intelligence and the precise logical entailment systems. Further researches revealed that F-logic is suitable to model not only well structured (eg. object-oriented) but also semi-structured information, such as that found in web pages [4].

2 Architecture

To get an overview easily, we present our solution by a concrete problem. The example, along with the implementation, serves also as a proof-of-concept.

Imagine that you are going to give a birthday present to a friend of you called Jim. You realise on Tuesday driving home after a hard day that the birthday party will be on Saturday and you have nothing to present him with. Fortunately, you know Jim has a new hobby: the contemporary English. But a problem arises: if you order a book tonight, you probably will not receive it in time. Therefore you have to personally go to a bookshop *nearby, open after usual working hours,* offering *high quality books on that topic at reasonable prices.* Is it possible to find such a shop easily now? Our proposal is a step towards it.

Obviously, we need the appropriate data sources and a mobile phone to deal with this problem. It is worth supposing two other components as it is on Fig. 1. To the databeses there are applications connected. Their task is to make the data available to other software operating in the same domain providing simplified functions, access control etc. The lion's share of the work is done by the agents. They mesh the pieces of information from the content providers and represent them in a uniform way. Agents serve as fat servers for the clients. The only job of the clients is to ask the agents. This is also adequate for current, immobile access. Since agents, not the clients at individuals, have the intelligence, this architecture is quite easy to manage.

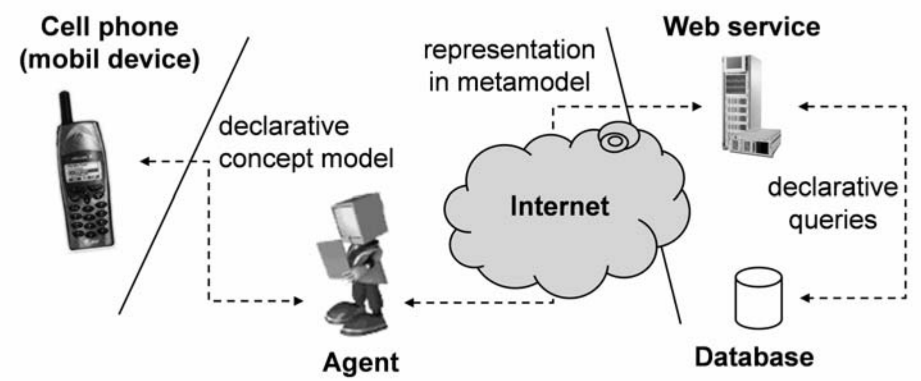

Fig. 1. Architecture for mobile access of deep web services

The architecture comprises three tiers if databases and the components that represent them, the web services are regarded as one unit. The only requirement for the databases is to answer schema queries as well as ordinary data queries. Based on this schema information the web service forms its *single, general* interface to the agents. F-logic can be successfully applied to this task because it makes no distinction between schema and data instances, it always treats schema as data and it allows the usage of object-oriented methodologies resulting in improved efficiency of development. In our example, the web service tells when each shop is open, provides geographical information and book records: their topic and price. But all these concepts are meaningful only for software designed for this subject. In general, a third party application is needed.

Web services are joined together via agents. That is why a single, self-descriptive interface is beneficial. The agents' world is made up of numerous pieces by means of information-retrieval methods. It is represented via well known semantic methods, eg. ontologies. In ontologies DAML+OIL is often used, which is based on description logic. Although expressive parts of DAML+OIL are undecidable[5,6], Description Frame Logic (DFL), an F-logic integrated with description logic is not[7], fortunately. Therefore, the application of F-logic turns out practical. In our example, the agent interprets the concepts *nearby, opening hours, usual working hours, high quality, reasonable price, topic* taking data from other databases and systems into consideration. Furthermore, if there is no proper book to buy, it can suggest another one based on less strict conditions. Note, F-logic and its roots in artificial intelligence can be good support for that.

Despite being thin, logic-based mobile clients have some benefit. On one hand, agents respond only to F-logic queries, which are easier to formulate by logic-based applications. On the other hand, the agents' world changes from time to time. To make use of that world, the model of the client software has to change, as well. It means that an application written in an interpreted language is necessary. Java, the most popular object-oriented language nowadays, falls into that category. So do object-oriented Prolog languages, in which, due to their higher abstraction level, problem solving requires much less lines of source code. That means eventually better debuggability, transparency and manageability.

We are currently implementing a system based on the introduced theory. We chose XSB Prolog as programming language for every component. XSB Prolog has an Oracle interface to declaratively communicate with the lowest elements of the architecture. A built-in implementation of F-logic, called FLORA, is also available. Unfortunately, there are no XSB Prolog interpreters on cell phones. So we use immobile computers instead of mobile devices but everything else is ready and works properly.

3 Conclusion

In this paper we addressed the problem of integrating content providers of the Web without duplication of functionality. We proposed a mobile architecture in which all components are logic-based. It incorporates the recent results and technology of object-oriented modelling, web services and the Semantic Web. The fact that the models of the components are closely related to each other not only requires less transformation at each interface but also simplifies the process. That has a favourable impact on productivity and cost-effective operation.

References

1. Michael K. Bergman: The Deep Web: Surfacing Hidden Value. Bright Planet White Paper (2001)
2. Hendler, J.: Agents and Semantic Web. IEEE Intelligent Systems **16(2)** (2001) 30–37
3. Kifer, Michael, Lausen, Georg, Wu, James: Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the ACM **42(4)** (1995) 741–843
4. Ludäscher, B., Himmeröder, R., Lausen, G., May, W., Schlepphorst, C.: Managing Semistructured Data with FLORID: A Deductive Object-Oriented Perspective. Elsevier-Pergamon Information Systems **23(8)** (1998) 1–25
5. Baader F., Calvanese, D., McGuinness, D., Nardi D., Patel-Schneider, P. (eds): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2003)
6. Horrocks, I.: Logical Foundations for the Semantic Web. Reasoning with Expressive Description Logics: Theory and Practice CADE-18 Invited Talk (2002)
7. Balaban, M.: The F-logic Approach for Description Languages. Annals of Mathematics and Artificial Intelligence **15(1)** (1995) 19–60