INTEGRATING PROPRIETARY ENVIRONMENTS AS REUSABLE COMPONENTS IN EDUCATIONAL SOFTWARE. AN EXPERIENCE WITH MATLAB, COM AND WEB SERVICES.

Emilio García Roselló¹, Jacinto González Dacosta¹, J. Baltasar García Pérez-Schofield², David Martínez Torres³, Manuel Pérez Cota¹

¹ Dept. de Informática, Universidad de Vigo, Edif. Fundición, Lagoas-Marcosende s/n , 36200 Vigo - España
² E.T.S. Informática de la Universidad de Vigo, Campus As Lagoas, s/n. 32005 Ourense- España.
³ Departamento de Electrónica y Computación, Universidad Tecnológica de la Mixteca Huajuapan de León, 69000 Oaxaca - México
e-mail: {erosello|jdacosta|jbgarcia|mpcota}@uvigo.es

Abstract

The field of educational software is currently involved in the rising of a paradigm of development based on reusable components or “learning objects”. So there are significant efforts directed to develop reusable educational software components. Some existing proprietary environments, which are “de facto” standards in particular educational domains, could be very interesting candidates to reusable components, given their functionality. But they are mostly offering very poor possibilities of integration in other software, owing to their dependency on a particular user interface, a too limited API, etc... In this paper a practical experience developing components to encapsulate and make reusable the MatLab environment is presented. The components, offering a true object-oriented API, are delivered as a local COM component as well as a distributed Web service employing SOAP technology. Several applications using these components are also presented, showing their potential reusability and usefulness in a wide range of domains and situations.

1. Introduction

Educational software has traditionally been based on authoring tools, generative reuse and “island applications” developed from the scratch to cope with relatively particular needs (Roschelle & Kaput, 1996; Roschelle et al., 1998). Therefore, educational software is fragmented, globally ineffective, and its development is too expensive and time-consuming (Roschelle & Kaput, 1996; Bork, 1995; García Roselló et al., 2002). These features led to a situation that several authors have described as a “pattern of failure” in the current paradigm of educational software development.

According to these authors, the adoption of a component-based paradigm in educational software community seems to be the most promising answer to this situation (Roschelle & Kaput, 1996). In this context, a new concept has recently arisen: learning objects. Learning objects are the basis of an emerging paradigm that has attracted the interest of many people and organizations such as LTSC, ARIADNE or IMS (ARIADNE, 2001; IMS, 2001; LTSC, 2001; Wiley, 2000; Singh, 2000; Barron, 2000; Friesen, 2001). According to LTSC, learning objects are defined as any entity, digital or non-digital, which can be used, reused or referenced during technology supported learning (LTSC, 2001).
Learning objects are the cornerstone of a new approach of computer-based instruction grounded in the object-oriented paradigm that promises a new way of conceiving, developing and delivering learning. Obviously, as far as software development is concerned, the interest is on digital entities, thus duplicable, deliverable and distributable on Internet (Wiley, 2000; Friesen, 2001). Moreover, in component-based development we will focus on reusable pieces of educational software that provide some functionality, someway available, preferably in Internet. So the adoption of this paradigm requires the availability of really reusable educational software components that allow developers to build applications composing, configuring and adapting prefabricated parts to their particular requirements.

Several educational projects have followed this approach with relative success (Educational Object Economy project, 2000; Educational Software Components of Tomorrow, 2001). However, they showed that collecting and offering a high number of components it’s not enough to increase reusability. On the contrary, the experience seems to point that high reusability in a particular domain usually depends, among other factors, of the existence of a small kernel of components that covers the basic functionality required for the applications in that domain (Roschelle et al., 2000; Szyperski, 1998; Biggerstaff, 1998).

The history of educational software offers a good analogy of this. Some proprietary software environments are practically “de facto” standards in certain educational domains thanks to offering, among other things, such kind of functionality. A good illustration of this is Matlab (Matlab, 2003) in the engineering education (see for example its regular presence in congresses or journals such as Wiley Periodicals Inc. (2002) or Frontiers in Education Clearing House (2003)). Other similar examples are Maple (2003) or Mathematica (2003) in mathematics.

So it seems evident that this kind of software can provide excellent candidates to reusable components. But having the required functionality is not enough for a component to be effectively reused, but other features are needed, such as encapsulation, a useful and friendly API, or independency of platforms, languages and user interfaces (Szyperski, 1998; Sametinger, 1997). Indeed all of those software environments fail in one or more of those requirements. Many of them are clearly designed to be used quite exclusively through their own user interface. Some of them offer a limited API, often complex, unfriendly or language-dependent. For example, Matlab offers several external API’s (Matlab Technical Documentation, 2003); but its C, C++ and Fortran API’s are clearly language (and even compiler) dependent. Also, its COM interface allows only the exchange of data of few types,
and the sending of text commands. In summary, they generally allow too limited possibilities of using them as components of other applications.

Therefore it could be interesting to provide components that encapsulate this kind of applications, offering a true object-oriented API, the way they could be effectively reused. A related aspect to consider is the fact that, besides providing components downloadable from the net for its local use, the interest is shifting more and more to offering distributed services (Liao et al., 2001; Wang et al., 2001; Linton & Solomon, 1999; Solomon & Struble, 2001).

When building software from reusable pieces that can be located on distributed systems, integration and interoperability are critical, in educational as well as in business applications. Thus standard platform-independent languages such as XML are fundamental (Singh, 2000; Liao et al., 2001; Wang et al., 2001) for intercommunication purposes.

Also, those Web services based on remote object invocation protocols such as SOAP, take shape as an interesting and still little explored approach to provide distributed learning objects on the net. Somewhat similar works have been developed in other areas, notably in mathematics (see for example Liao et al. (2001); Wang et al., (2001), Linton & Solomon (1999), Solomon & Struble (2001). But instead of using a general purpose remote object invocation protocol such as SOAP, they choose to use domain-specific XML-based languages like OpenMath (Linton & Solomon, 1999) or specialized API’s (Liao et al., 2001). They also generally focus in providing a command-based interface, so the developers see them more as “command-line processors” than true object-oriented software components.

In this paper we describe our experience in building a locally reusable component, as well as a Web service, which encapsulates the functionality of Matlab, principally thought up to be reused in educational software development.

The aim of our work was to allow an easy integration of Matlab, as a component, in other applications. We also aimed to fit the component with a simple and easy-to-use object-oriented API that allows developers to handle items (variables, functions and results) as objects (in the object-oriented programming sense). This way, we intend the component could be really seen and used as a reusable software component.

The rest of this paper is organized as follows: firstly, the development of the component and the Web service is discussed. Then, some different applications, developed to test the usefulness and simplicity of employment of the component and the Web service for educational software building, are described. Finally we explain the conclusions and further work.
2. Development of the MatlabCOM component

In order to encapsulate the functionality of Matlab in a reusable component, we built a COM library, which we named MatlabCOM. The design of the component was done keeping two objectives in mind: the first one was to provide a true object-oriented interface to Matlab which could be effectively used for local applications development. The second one, was to reuse the same component to implement the Web service.

![Figure 1. UML diagram of the MatlabCOM component](image)

The hierarchy of interfaces it exposes are shown in figure 1. The IMatlabSession interface encapsulates a session. As most calculations aren’t generally intended to be done in a single step, the session allows a client application to define variables and perform several operations in its own workspace (Liao et al., 2001; Wang et al., 2001). The IMatlabVar interface and its descendants encapsulate different types of variables. And finally the MatlabFunc interface allows to call a function and to get its result as an IMatlabVar object.

This schema, even though relatively simple, allows to manipulate variables and execute any function in Matlab engine, without having to deal with commands or strings, but objects created and managed as any other object of the programming language used to develop a given application (it is implied that the programming language employed must have object-oriented capabilities).

3. Development of the MatlabSOAP Web service

Once MatlabCOM component available, the development of the Web service, which we named MatlabSOAP, was reasonably simple. Since we intended the service to provide the same interface as the local component, the immediate solution was to build a proxy component which simply acts as a bridge between SOAP clients and the COM component. But there was a problem due to the nature of SOAP: server objects are created on request to
resolve a given task, and destroyed immediately after. Therefore it doesn’t support a concept such as a session, needed for non trivial calculus, (in order to define variables … etc.), which implies persistence between requests. In order to solve this limitation, a component providing this session mechanism (through simple persistence) had to be added. We implemented it as an OLE automation server. Once started, it keeps always running and serves the requests from the SOAP module, which stands as a simple proxy of client’s requests. When a client requests for opening a new session, the SOAP module transmits this request to the automation server, which creates the new session and returns an id to the SOAP module, which in turn returns it to the client. Thus, the session remains opened in the server until it is closed by the client. Any subsequent requests of the client have to include the session id of the session in which they are going to be executed. The resulting architecture is shown in figure 2.

![Figure 2. Architecture of a client/server application based on the MatlabSOAP service](image)

4. Development of educational software applications using MatlabCOM and MatlabSOAP

Several applications were developed using the COM component and the Web service previously described, testing, this way, their usefulness and simplicity of use in educational software development. Below we briefly describe some of the most significant ones.

4.1. Simple integration in Microsoft Office suite

One of the applications was a simple module encapsulating calls to the MatlabSOAP service through simple functions built in Visual Basic. This is interesting for educational purposes, as for example, this allows calling Matlab functions from Excel, an application very used in education, as well as from any other application of the Microsoft Office suite. This can be done by means of some added functions that can be used in the same way that Excel built-in functions are employed in a worksheet. The ExcelLink add-in included in the Matlab package
offers quite the same functionality, but, obviously, it requires Matlab to be installed on the local machine and it only works in Excel.

4.2. Integration of Matlab function library in Model-Lab software environment

The integration of the MatlabCOM component in an ecological modelling software, Model-Lab (García Roselló et al., 1998; García Roselló et al., 2003b), a previous project carried out by our research group, was also developed. Model-Lab allows to easily add new DLL’s (Dynamically Linked Library) containing functions without having to recompile it, but simply putting in a file the data about the DLL’s and functions they export. So we only had to develop a DLL acting as a proxy. This DLL exports the desired functions, so they can be used from Model-Lab, but its implementation consists of a few calls to the MatlabCOM component. So it supposed a very simple way to achieve a high increase in power for Model-Lab, extending it with a very comprehensive library of functions and enhanced calculus capacity (figure 3).

4.3. Enhancement of Neuro-Lab software environment with Matlab Neural Networks Toolbox

MatlabCOM was also used to integrate the functionality of the Matlab Neural Networks Toolbox (MNNT from now on) in another previous project, an educational software already developed by our team, called Neuro-Lab (García Roselló et al., 2003). We wanted to enhance this software so it could support more types of artificial neural networks (ANN’s) and training algorithms, but keeping its visual user interface, specially designed for educational purpose. So instead of having to implement known but complex algorithms and data structures, the employment of the MatlabCOM component could offer us the required
functionality with a lower development cost. Another important added value of this solution was the common reliability, correctness and robustness of these toolboxes. However, the direct use of the MNNT through MatlabCOM isn’t particularly easy, as we had to deal with several complex functions and data structures. So, in order to achieve an easier integration and reuse of the functionality of this toolbox from other applications, we decided to develop a new domain-specific component which we named MatlabANNCOM. This component uses the generic services provided by MatlabCOM to access MNNT functionality, offering in turn an object-oriented API specifically designed to work with ANN’s. This component defines a hierarchy of classes which implement different kinds of ANN’s and training algorithms in an easier and more intuitive way (following the design proposed in García Roselló et al. (2003)), hiding the complexity of the MNNT. The architecture of the system is shown in figure 4.

A version of this environment which uses the distributed MatlabSOAP Web service was parallelly developed and tested, as migrating from MatlabCOM to MatlabSOAP is quite trivial.

![Figure 4. Architecture of the Neuro-Lab environment.](image)

### 4.4. A simple Web-Calculator with MatlabSOAP

Applications developed using Web-oriented languages and tools are currently an usual way of delivering educational software. That’s why we considered interesting to build an application using a script language such as PHP to verify the easiness of using our Web service in this kind of software solutions. The application simply allows to create and manipulate variables
and to use functions of Matlab (Figure 5). The results were very satisfactory, as there were no particular difficulties thanks to the SOAP support provided by the NuSOAP module (Ayala, 2003).

5. Conclusions

Matlab, like some other proprietary software environments, has a large history of successful educational usage in its domain, in part because it covers quite well the basic functionality of its particular domain. But its limited integrability as a component to build other applications hinders the natural interest in reusing it.

In this paper we presented a practical experience in developing components to encapsulate Matlab. The main objective was to allow reusing the functionality of Matlab in other educational software applications, providing components with a true object-oriented API, also available on widely used component platforms, as COM and SOAP. Several applications which reused these components were also presented, showing their usefulness for educational software development.

Local components platforms and distributed Web services can provide a powerful way to build applications from platform-independent components. This is a very interesting feature in the educational software field, where there is a clear trend towards a component-based paradigm, as well as towards the Internet as the favourite platform to deliver software.
6. Further work

The experience in integrating MatlabCOM in Neuro-Lab, previously described, showed us that the genericity of the functionality of a component like MatlabCOM can limit its reusability for very specific domains. Currently we are working in new, related, projects, which follow the path we opened with the solution presented here. We intend to see the viability and usefulness of developing n-tiered distributed architectures of educational Web services, where a domain-specific service is implemented over domain-generic services of lower levels. At the lowest level, a Web service like MatlabSOAP would support the basic functionality of the domain (figure 6).

In fact, environments such as Matlab implicitly apply a similar rule, allowing to add new functions by means of modules or libraries (toolboxes in the case of Matlab) to provide domain-particular functionality.

As one of the topmost used features of Matlab is its ability and versatility for doing graphs, we are currently developing an extension of the API of MatlabCOM and MatlabSOAP to allow client applications to create, manipulate and import to their own environment graphs created in Matlab.

![Figure 6. An N-tiered architecture of educational web services basing on MatlabSOAP service.](image)

7. References


