

COMPONENT SOFTWARE FOR APPLICATIONS IN REHABILITATION ROBOTICS

John L Dallaway

Department of Engineering, University of Cambridge, UK

Abstract

Object-enabling system software facilitates the construction of complete software solutions using prefabricated executable components. The potential benefits of this technology in the field of Rehabilitation Robotics are discussed. Specific features of Microsoft OLE are introduced and a technology demonstrator based on this software is described. The demonstrator makes extensive use of existing software components to enhance functionality and reduce development costs.

Background

The diversity of research and technical development in the field of Rehabilitation Robotics has discouraged the re-use of software seen within other application areas. Previous initiatives to provide generic software solutions have experienced some success, but it remains common practice to generate bespoke application software for robot command purposes.

The potential benefits of code re-use in this field are compelling. Researchers are able to concentrate on the specific issues under investigation. Labour costs and development time are reduced while system reliability is enhanced by

the incorporation of pre-tested sub-systems. Project continuity and the potential for collaboration are also enhanced [1].

The key to code re-use lies in effective modularity. An increase in the use of object-oriented programming techniques has generated the potential for such modularity. However, these techniques, of themselves, have been insufficient to stimulate the widespread adoption of modular application software.

A more recent form of object technology known as *object-enabling system software* now facilitates the construction of complete software solutions using prefabricated executable components.

Component software

An object is a self-contained software module which possesses a set of properties and a set of methods by which the properties may be manipulated. The term *software component* refers to an executable object which is designed to work in conjunction with components from other sources to form software solutions customised to precise user requirements. The rules by which software components interact are

defined by the underlying object-enabling system software.

Microsoft OLE is an example of object-enabling system software which provides a set of object services for the Microsoft Windows and Apple Macintosh platforms [2].

Implementations for major versions of Unix are also in preparation. OLE services are extensive and generally referred to by category. Categories of particular interest in this discussion are OLE Automation and ActiveX Controls.

OLE Automation is a mechanism by which application software may expose objects. In this way, an application is able to offer services to other applications using well defined software interfaces. The services may execute either as part of the client process or as a separate server process.

ActiveX Controls are executable software components which may be incorporated into applications to extend their functionality. These components may provide a novel interactional element within a user interface or handle data of a particular format.

Modular robot command software

Potential uses for component software may be identified by considering robot command systems in terms of logical objects. The function and structure of each application-level object is commonly influenced by the system-level objects with which they are constructed. For example, a software

component which implements a world model for a robot system would inherit structure from system-level database access objects.

The cross-platform interoperability of certain object-enabling system software facilitates the distributed implementation of application software across a heterogeneous computer network. Such flexibility could prove invaluable in a research context. For example, an experimental robot vision sub-system could be hosted on a Unix-based workstation, while the user interface is developed on the intended end-user platform.

An implementation

A component software technology demonstrator is being developed in the context of ongoing research into rehabilitation robot command. It is envisaged that the design and implementation of such a demonstrator will lead to an enhanced understanding of the logical components of robot command software systems. The demonstrator is based on Microsoft OLE technology and will be coded using a number of programming environments including Visual C++ and Visual Basic. It will also incorporate existing software components where practical.

To date, an advanced software teach pendant and world model database management system (DBMS) have been designed. These components have been demonstrated in conjunction with an

RTP protocol manager and an IMP8-upgraded RTX robot arm.

The world model DBMS is implemented as an in-process OLE Automation server. It uses Data Access Objects (DAO) to interact with the Microsoft Jet database engine. The server enforces a hierarchical structure for the world model and implements methods for creating and deleting records and for the manipulation of relative robot location information. Low-level mathematical functions are contained within a dynamic link library coded using Visual C++.

The teach pendant is implemented using Visual Basic. It controls the world model DBMS using OLE Automation. World model data is manipulated using ActiveX masked edit controls bound to specific database fields. The database hierarchy is presented in the form of a tree using an ActiveX TreeView control.

The teach pendant communicates with

the robot via a device-independent command shell object using OLE Automation. The shell forwards commands to the RTP protocol manager in a device-specific manner. Its presence facilitates the rapid integration of higher level demonstrator components with alternative robot hardware. Low-level communication functions are contained within a dynamic link library coded using Visual C++. The interaction of software components within the demonstrator is illustrated in figure 1. Novel software components, implemented in the context of this work, are shown in bold type. Existing components are shown in normal type.

Future development of the demonstrator will concentrate on user interaction. An existing iconic interface [3], developed for office-based robot workstations, will be implemented as an ActiveX control and the integration of a 6 degree of freedom force sensitive input device is

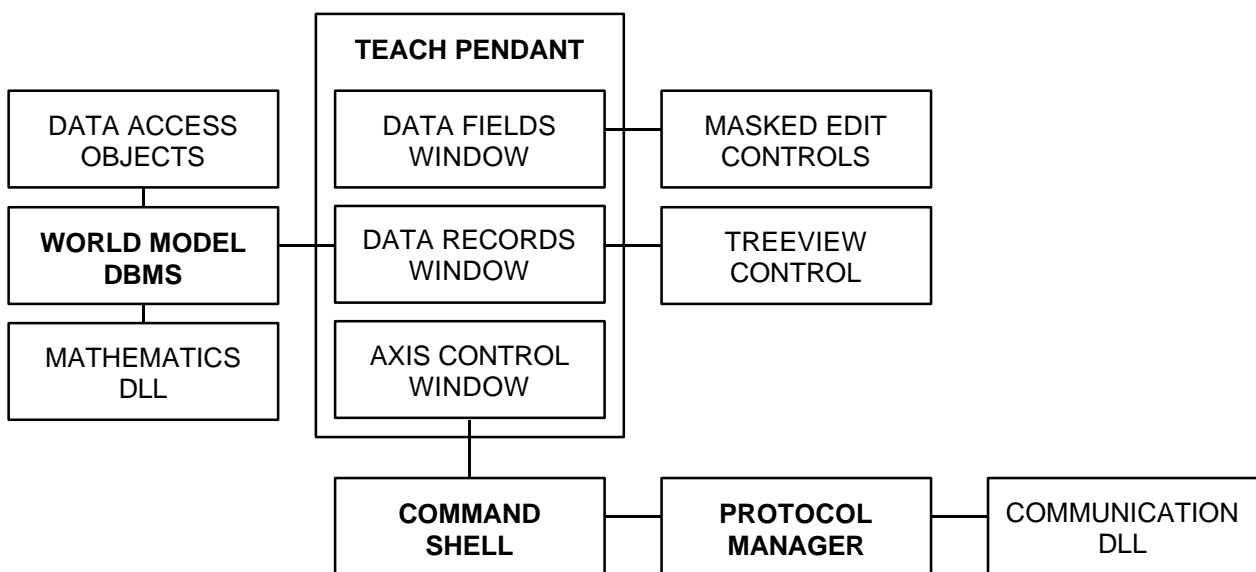


Figure 1. Structure of the component software technology demonstrator

planned. The feasibility of a novel scripting component, building on experience with the Cambridge University Robot Language (CURL) [4], is also under investigation. Preliminary work towards a complete object-based model of a generic rehabilitation robot command system has now commenced.

Conclusions

Object-enabling system software provides an opportunity to generate flexible reusable software components for applications in rehabilitation robotics. This technology enables developers to build novel robot command applications which make substantial use of existing third-party code.

Developers should consider the benefits of object technology within their own development programmes. Coding effort is reduced, reliability is enhanced and their exists scope for the commercialisation of novel software components.

References

[1] Gilbert, M.L., Caruso, J.P., Mahoney, R.M., Fee, J.W., "Development of a programming environment for rehabilitation robotics", Proceedings of the RESNA 89 Annual Conference, New Orleans, 377-378, 1989.

[2] Brockschmidt, K., Inside OLE – Second Edition, Microsoft Press, Redmond, 1995.

[3] Dallaway, J.L., "Human-computer interaction within robotic workstations", Proceedings of the RESNA 96 Annual Conference, Salt Lake City, 339-341, 1996.

[4] Dallaway, J.L., Mahoney, R.M., Jackson, R.D., Gosine, R.G., "An interactive robot control environment for rehabilitation applications", *Robotica*, 11, 541-551, 1993.

Acknowledgements

The author acknowledges the support of the Leverhulme Trust (ref F/452/B).

Address

Dr JL Dallaway
Department of Engineering
University of Cambridge
Trumpington Street
CAMBRIDGE
CB2 1PZ
United Kingdom

Tel: +44 1223 332600
Fax: +44 1223 332662
E-mail: jld@eng.cam.ac.uk