Session 3663

Using Object Linking and Embedding (OLE) as an Integration Tool

Troy E. Kostek
Purdue University

Introduction

Today, OLE (object linking and embedding) technology can be used to integrate an entire manufacturing operation ranging from the factory floor to the organization's information and management systems. Industrial OLE-based component products (such as Rockwell Software's RSTools™), combined with a Microsoft Windows development tool (Visual Basic, for instance), are providing many benefits to manufacturing engineers and systems integrators. In some applications, component products are replacing traditional Man-Machine Interface (MMI) software packages[6].

OLE technology is easy to use and increases the software developer's productivity. With the help of OLE technology, novice programmers are now capable of writing powerful programs that only computer science majors were capable of writing a few short years ago. If this technology is properly taught to students pursuing a degree in manufacturing engineering or manufacturing technology, the graduates of these programs will be capable of contributing to a company's manufacturing integration efforts in a significant way and thus greatly benefit their employers.

The purpose of this paper is to examine the benefits of using OLE-based component products and to provide an example of a leading edge commercially available product which incorporates OLE technology. The paper will also suggest how this technology can be used in a manufacturing curriculum.

I. Introduction To OLE Technology

OLE enables two or more applications to work together and share data. OLE technology (first released by Microsoft in 1991) was originally an evolution of DDE (dynamic data exchange). DDE is a messaging system which allows two Microsoft Windows applications to share data. Compared to DDE, the average computer user will find OLE technology relatively easier to implement (for additional information on DDE and how it can be used to teach manufacturing integration, see reference 1).

Many people associate OLE with applications such as spreadsheet data being embedded or linked into a word processing document. In this application, changes can then be made to the embedded (or linked) spreadsheet data by double clicking on the data. OLE automation, however, continues to evolve at a rapid pace. Today, OLE-based technology is heavily used by Microsoft operating systems (Windows 95 and Windows NT), Windows development tools such as Microsoft's Visual Basic, Internet/Intranet Web browsers, and industrial software companies such as Rockwell Software Inc.

To better understand some fundamental OLE terminology, consider an example in which cells A1:C7 of a spreadsheet are embedded in a word processing document. Now consider the following terminology[7]:

> Object - The data supplied by another application. It is a piece of data that you move from one application to another in it's native format. In the above example, the cell range A1:C7 is the object.
>
> Client - The application that contains or holds the linked or embedded object. With respect to the above example, the client is the word processing document. The client is also called the destination and in Visual Basic version 4 it is called the OLE controlling application.
>
> Server - The server is the application which defines and exposes OLE objects. The server is the application that the client calls when it is necessary to make changes to the embedded or linked object. Note that the embedded or linked object is still in the native format of the application that was used to create it. In our example, the server is the spreadsheet program. The server is also called the source and in Visual Basic version 4 it is called the OLE object application.
>
> Compound document - This is the document that contains one or more objects. In the above example, the word processor is a compound document.

The basic difference between linking and embedding is: (1) where the actual data is stored, and (2) what applications have the ability to access and change the data.

II. OLE Custom Controls

A. Introduction

OLE custom controls are part of the ongoing evolution of OLE technology. An OLE custom control is a relatively small software package which serves a specific function and is reusable in the sense that it can be used over and over in order to develop custom Windows applications. In the Windows environment, OLE custom controls (OLE servers) are graphically "dropped" into an OLE client application. The OLE client (also called a container) can be any Windows application or software development tool that supports OLE custom controls. Commonly used software development tools include: Visual Basic, PowerBuilder, Delphi, CA Realizer, Rockwell Software's RSView32, and Microsoft's Visual C++.

OLE custom controls are pre-built pieces of functionality which can perform general tasks such as graphics, charting, mailing, and faxing. OLE custom controls are also available from Industrial software companies such as Rockwell Software Inc. Industrial OLE custom controls can be used for:

- Acquiring data from a PLC (programmable logic controller) or other field device
- Processing data acquired from a PLC

- Displaying PLC data via a custom designed graphical user interface (GUI)
- Connecting and logging data to databases
- Fast and reliable data communications

OLE custom controls are the successor to the Visual Basic VBX (Visual Basic Extension). The VBX allowed Visual Basic programmers to extend the built-in capabilities of the Visual Basic programming language. Although the VBX concept was popular, the VBX had the following limitations:

- The VBX was designed to work with Microsoft's Visual Basic. In this sense, the VBX is closed or proprietary since it is not compatible with other Windows development tools.
- The VBX is limited to 16-bit Windows applications.

OLE custom controls are open since they use standard OLE technology which is an open standard and thus is supported by many vendors. An OLE control can be "dropped" into any Windows application or Windows development tool that supports OLE custom controls. The goal is to be able to use the same OLE custom control across multiple hardware platforms, operating systems, and development environments. Today OLE controls are only supported by Windows 3.x, Windows 95, and Windows NT. Support for OLE technology on popular UNIX platforms has been announced. OLE custom controls can be used in both 16 or 32-bit Windows applications.

B. OLE Changing Terminology

As OLE technology evolves, it's associated terminology changes accordingly. As mentioned above, OLE custom controls are the successor to the Visual Basic VBX (Visual Basic Extension). OLE controls are sometimes called OCX's because they use "OCX" as a file name extension. However, the latest term used to describe OLE custom controls is the term ActiveX. An ActiveX control is essentially the next version of OLE custom controls. Microsoft recently renamed OLE controls to ActiveX controls as part of the evolutionary change of OLE technology. ActiveX controls are OLE custom controls which have been extended to embrace the Internet.

Application development in the Visual Basic programming environment has become a standard for what is called "component-based development." The term component is used to describe an ActiveX control which can be placed in any compatible ActiveX container such as Microsoft Visual Basic, Access, or Internet Explorer. The goal is that different software companies will write manufacturing related component products and manufacturing engineers and systems integrators will then be able to select the best components from multiple vendors.

C. Benefits Of OLE Custom Controls

The following general benefits are realized when using today's OLE technology:

Rapid Development - OLE controls offer canned or turn-key functionality that would otherwise have to be developed. Program development is a simple object-oriented process. Development time is reduced because less code is written and therefore fewer programming bugs must be diagnosed and repaired.

Flexibility - OLE controls can be used with a wide variety of Windows development tools and Windows applications. By changing their properties at design time or run time, OLE controls can be customized to meet one's needs.

Ease Of Use - OLE controls are graphical and object oriented. A graphical approach generally promotes ease of use. When using object oriented development tools, developers can focus more on the "big picture." Since the objects perform specific tasks, the programmer does not have to write the program from scratch and thus "wrestle" with a multitude of details. Novice programmers now have capabilities previously available only to advanced programmers.

Reduced Cost For End User - As mentioned above, development time is reduced since less code has to be generated and debugged. A reduction in development time generally decreases the cost of the project. And the developer using reusable OLE-based component products only uses the components which are necessary to complete the task at hand. The end customer is not buying a traditional MMI package which is loaded with many features that may not be applicable for their application.

III. Rockwell Software's Component Products

A. Introduction

Rockwell Software Inc. (RSI) is the first industrial software company to introduce a comprehensive suite of ActiveX controls geared towards the industrial market. RSI's industrial component suit consists of multiple ActiveX controls and several development add-ins for Microsoft's Visual Basic. These OLE-based tools are well suited for applications that must acquire and graphically display plant floor data, provide alarming capabilities, and log data to a database. In short, these products provide a cost effective link from the factory floor to the top floor information and management systems and the ability to create MMI interfaces. These component products can be used to develop applications for discrete manufacturing, process controls, and SCADA (supervisory control and data acquisition) applications. RSI's component products were designed to be used in a 32-bit environment. Thus, one should use Microsoft Windows® 95 or Windows NT®. Microsoft Visual Basic version 4.0 or any other Windows development tool capable of utilizing ActiveX controls.

B. Features

RSI's component products offer the general benefits associated with OLE custom controls as previously discussed, as well as high speed PC-to-PLC communications, extensive database support, and Internet access.

RSI's OLE-based component products use DDE technology to communicate to shop floor devices such as PLCs. Instead of using the DDE protocols built into Visual Basic (which were not designed to support robust industrial applications), RSI uses their DDE protocol called

AdvanceDDE™. RSI's AdvanceDDE protocol was designed to support reliable, high volume, high speed PC-to-PLC communications. It is interesting to note that the popularity of OLE-based technology is not completely replacing DDE. Using a DDE server (such as RSI's WinLinx), RSI's ActiveX controls use DDE to communicate to the devices found on the shop floor. The difference in this case is that the systems integrator does not have to understand DDE, that is, OLE technology is making DDE transparent to the programmer.

RSI's component products provide high levels of database connectivity. First, realize that Visual Basic can attach to and share data with any OBDC (Open Database Connectivity) compliant database. Examples of databases which support the OBDC standard include: Access, Paradox, dBASE, FoxPro, SQL Server, Oracle, and others (see reference 2 for additional information on how to use Visual Basic's data control). In addition, each of RSI's component products are "data aware" which means they can directly access, display and make changes to data that is stored in any OBDC-compliant database without the need for writing any code. Each component product can be configured to automatically log data to an OBDC-compliant database on a time-driven basis or whenever the relevant data changes.

ActiveX controls are Internet/Intranet enabled. RSI's product components can reside in any 32-bit Web browser that supports ActiveX controls (such as Microsoft's Internet Explorer). A home page can be set up to examine shop floor data over the Internet in order to promote "inter-company" sharing of data.

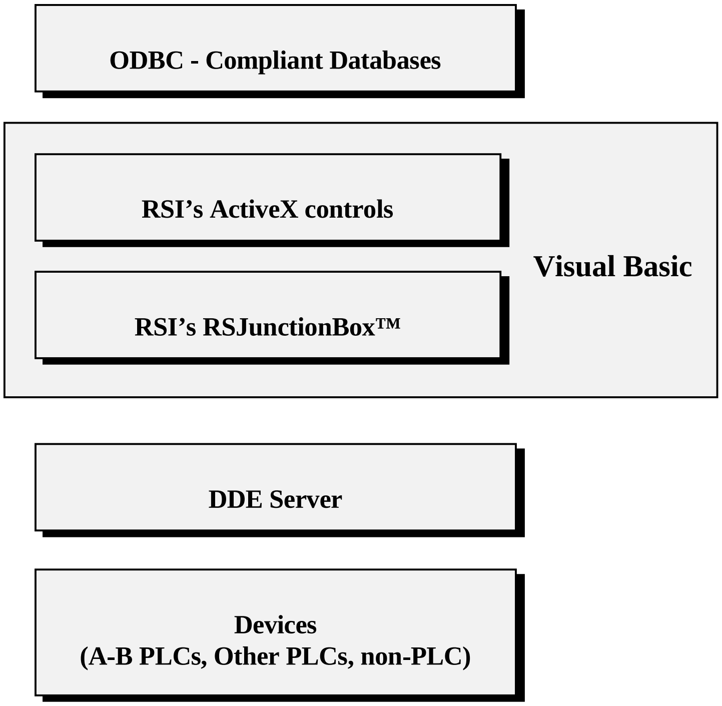
C. Basic Architecture

RSI's component products consist of RSTools™ (7 ActiveX controls described below), RSJunctionBox™, RSPowerTools™ (3 more ActiveX controls), RSWorkbench™ (6 Visual Basic add-ins), and several other ActiveX controls which can be purchased separately. To better understand RSI's component products it helps to examine the overall system architecture as illustrated in Figure 1.

The RSJunctionBox is a communications module which allows RSI's ActiveX controls to communicate with DDE servers which support the AdvanceDDE protocol. RSJunctionBox serves as a "traffic cop" between the ActiveX controls and the DDE server and adds AdvanceDDE capabilities to Visual Basic. In general, the DDE server understands how to talk one or more of the various DDE protocols and equally important it understands how to communicate to a specific shop floor device. Specifically, RSI's DDE Servers such as WinLinx and RSLinx support several DDE protocols including AdvanceDDE (the concept of a DDE server is further discussed in reference 1).

When software developers create applications using Visual Basic and RSI's ActiveX controls, their software products are not subject to any run-time license fees. Each run-time station, however, will have to have RSJunctionBox and a DDE server. The developer buys RSTools only once.

Figure 1 - Basic Architecture



D. The 7 Components In RSTools

The following RSI component products will work with Visual Basic or any other development environment which supports ActiveX controls[3]:

RSData - This ActiveX control provides numerical display of device data. The RSData control basically emulates a digital panel meter. RSData can also be used to log data to a database whenever new data is available (the data changes) or on a time-driven basis.

RSVessel - This control is used to show the current level of a particular parameter. The control can be used to graphically represent storage tanks, vats (or any other industrial device/shape) and display their level by flood-filling the graphical object based on the current real-time value of the parameter being monitored. Bit map backgrounds are also supported. Several fill modes are available (such as: top-to-bottom, bottom-to-top, mirror-out, and area fill).

RSWheel - This control emulates a traditional thumbwheel switch and thus allows input of numerical data. The control can also be used to display values as well. The RSWheel control supports both read-write or read-only modes. Data can be displayed using decimal, octal, binary, or hexadecimal.

RSGauge - This control serves as a graphical gauge which can be used to monitor any value stored in the PLC's memory. Options include round, semi-circular, arced, or linear gauges. Dual scales can be displayed if desired (e.g., angular distance in degrees or radians). RSGauge can also alert an operator when a value enters a previously defined warning zone or if the value is not within the defined min/max range.

RSButton - This control emulates an industrial push-button. Options include momentary, toggle, or rocker type buttons. RSButton can also be used to emulate a pilot light. Developers can make the buttons various sizes, shapes, and colors. A user provided bitmap drawing can be inserted on top of the control.

RSSlider - The RSSlider control provides highly customizable data entry. Features include min/max value restriction and a dual scale display (e.g., simultaneously displaying degrees C and degrees F). The current value of the variable can be displayed on the slider.

RSCompare - This is a multi-state evaluation control which can be used to compare the current value of the control with a predetermined setpoint ("Absolute Compare Method") or compare the current value with the previous value ("Linear Compare Method"). RSCompare graphically represents above, below, and equal conditions.

E. Additional Controls

Several other ActiveX controls and Visual Basic add-ins are available from RSI and this list will surely continue to grow. Some of the additional ActiveX controls are briefly mentioned below[3]:

RSAlarm - Is a graphical alarm management tool which is data aware (i.e., alarm data can be logged to a database without having to write any code.). Its now easy to add full-featured alarming capabilities to your Visual Basic applications.

RSAnimator - Is used to create custom designed screen animations.

RSEventMaster - Perform complex event handling and notification without writing extensive code. Integrates directly with RSAlarm.

RSLadder - Provides ladder logic viewing (online and offline), diagnostics, search capability, and more. Now you can add ladder logic viewing capability to your own programs or any other application which supports ActiveX controls (e.g., Internet Explorer).

RSChart - Performs real-time graphical charting. Provides automatic data logging to databases. Supports over 50 different 2D and 3D chart types.

IV. Using RSTools In A Manufacturing Program

Any manufacturing curriculum concerned about computer literacy should teach their students the fundamentals of OLE technology. Basic OLE competence can be taught simply by using some of the accessories (such as Paint and WordPad) built into Windows 95[8]. Curricula designed to teach students the fundamentals of manufacturing systems integration should focus heavily on both OLE and DDE. By itself, Visual Basic 4.0 is an excellent tool to teach the fundamentals of

OLE and DDE. The combination of Microsoft Visual Basic and Excel, can certainly provide a wide breadth of OLE/DDE knowledge.

An industrial package such as RSI's RSTools can and should be used in a wide range manufacturing courses. The table shown below illustrates how each of the tools previously discussed could be strategically introduced into a manufacturing curriculum.

Table 1 - Teaching RSTools

A course which focuses on:	**Could use the following tools:**	**Notes:**
Introduction to PLCs	RSLadder, RSData, RSButton, and RSWheel.	Ladder logic viewing is a must. Simple data acquisition
Introduction to Databases	RSData	Show how VB's data control works and show how RSData is data aware
Data communications & LANs	RSData	Teach fundamentals of OLE and DDE Teach PC-to-PLC communications Introduce Internet-based data collection
Sensors and Data Acquisition	RSData, RSGauge, RSVessel, RSCompare, RSChart, RSAlarm, RSSlider, RSWheel	Practically all of the RSI ActiveX controls are data acquisition tools. Show how to create a graphical MMI with alarms.

V. Conclusion

OLE is an important technology which is constantly evolving. Commercially available OLE-based tools specifically designed for manufacturing environments are available today. All systems integrators working in a Microsoft Windows environment can save themselves time, make their job easier, and can reduce development costs by using OLE-based component products such as RSI's RSTools. As manufacturing educators, it is our responsibility to make sure our graduates are introduced to this leading edge technology.

References

1. Kostek, T. (1995). Using Dynamic Data Exchange (DDE) as an Integration Tool. ASEE National Conference. Washington, DC.
2. Kostek, T. (1995). The Integration of Database Technology into an Existing Manufacturing Curriculum. ASEE North Central Section Annual Conference. Columbus, OH.
3. Rockwell Software Incorporated. (1996). Industrial Components. Product Literature.
4. Cash, S. (1996). Visual Basic Programming For The Industrial Environment. Allen-Bradley Journal. Chicago, IL: Putman Publishing.
5. Harding, V. (1996). OLE for Scada, Process Monitoring, and Control Applications. Allen-Bradley Journal. Chicago, IL: Putman Publishing.
6. Hamilton, J. (1996). OLE and The Software Revolution. Allen-Bradley Journal. Chicago, IL: Putman Publishing.
7. Norton, P. (1995). Peter Norton's Guide to Visual Basic 4 For Windows 95. Indianapolis, IN: Sams Publishing.
8. Norton, P. (1995). Peter Norton's Complete Guide to Windows 95. Indianapolis, IN: Sams Publishing.

TROY E. KOSTEK
Troy E. Kostek is an Associate Professor of Mechanical Engineering Technology at Purdue University. Since 1986, he has been teaching several senior-level courses in the Computer-Integrated Manufacturing Technology program at Purdue. His main areas of interest include programmable controllers, sensors, and machine vision.