The New Office Suites: A Developer's View

Here's a quick start into choosing an Office package for your development needs.

By Basil McDonnell

By now, you're starting to recognize that the "Office Suites" are more than just a collection of programs. Soon, when you're asked what your favorite development environment is, your answer may not be C++, PowerBuilder, or CA-Clipper. You may answer Microsoft Office, Lotus SmartSuite, or the Perfect Office Suite.

This article discusses the suites from a developer's point of view. The intent is to provide you with some guidelines to help you decide which is best for you—and your users.

How to choose

If you're lucky, you'll have the opportunity to participate in your organization's suite selection process. I say "lucky," because once the suite is bought and installed, you're going to have to live with it. Once it's on hundreds, perhaps thousands of desktops, it's likely to be there for years—for better or worse. And you can bet that someday you'll be asked to program in its environment. So take a stand early, and don't leave the decision strictly to the purchasing department.

There are two schools of thought when it comes to choosing a suite. One holds that the suites are all about the same, and that they constantly leapfrog each other, so the one that's worst this year will leap past the others in the next year or so. My experience with the three suites has put me in the other school, which regards this assumption as a huge mistake. If this principle were valid, Visicalc would still be on the market. There are huge differences between the suites from the developer's perspective, and the differences reach to the heart of the suites, down to the most basic level of their structure. There's strong reason to argue that certain features or capabilities, for some of the suites, fall into the "you can't get there from here" category—those capabilites that will never show up in the package. From a developer's point of view, it's no use saying "ours will be just as good next year" when the development project has to start in a week!

One developer's wish list

What would the perfect suite include? The basic requirements, as I've identified them, are: a common macro language; a consistent programming IDE; an open toolbox into the applications; wide-ranging data links; full OLE support; and terrific developer support. Let's look at each of the criteria.

A common macro language

All suites began life as loose collections of products, and each incorporated some kind of macro language. Unfortunately, the languages in the individual products usually had little or nothing in common with the others. When combined, the various programming languages become an expensive proposition, since the organization has to train and maintain experts in not one, not two, but sometimes three or more programming languages, all supposedly part of the same application development environment.

It's not just an unhappy situation, it's unworkable. One of the most important problems in multiple-application development is deciding which application program will host the overall system. Will it be the word processor calling on the resources of the spreadsheet, or the other way around? With a mix of programming languages, if you take the wrong route, you're stuck. With a common language, all you have to do is cut and paste.

Developers and programmers tend to work strictly in the application that supports the language they're most comfortable with. The result is often apparent in applications written under today's current suite offerings. You end up with applications that are clearly database-oriented hosted in Excel, what-if applications hosted in WordPerfect, and so on. With a common macro language, developers throw off the bounds of the host application. It's the freedom to move, and it's a minimal demand.

Consistent programming IDE

A common macro language is one step to programmer happiness; the second step is making that common language work the same way, regardless of which application hosts the language. This means the editor works the same way, programming elements work the same way under each language, compiling works the same way, and the same debugging tools are available.

This way, the benefits of the common macro language aren't lost in problems related to implementation under different applications.

Continued

Basil McDonnell is a Vancouver-based software developer and consultant specializing in PowerBuilder and Xbase programming. CompuServe 74007,2046.

An open toolbox

One of the key advantages to programming in a suite should be the ability to program capabilities of the suite into your applications. For example, if the word processor includes a boolean search dialog, you should be able to get at the functions that make it run. You should be able to do a boolean search in your dialog. Anything the application can do, programmers should be able to include in their applications—and with relative ease.

OLE

Object linking and embedding (OLE) is the Microsoft way of patching documents together from different Windows applications. From simple beginnings, OLE has expanded into a major structural technology, one that does all kinds of jobs. For developers, it's a way to host your system in one application and embed resources from other applications into your suite. With OLE 1.0, you could embed and link objects; with OLE 2.0, you can program the objects. The embedded object can be manipulated by the host application in the host application's programming language.

Data links

Stitching together corporate applications eventually involves a data table, and grabbing some of the data for one of the other applications. As developer's toolkits, suites have to be able to reach out and touch a lot of different data sources. From a programming point of view, it's vital that there be a single way of retrieving data and it has to be easy to switch data sources.

When you're talking about easily switching between data sources, you're usually talking SQL. The corporate plan these days is likely to include a number of different types of SQL databases, and your application is going to have to talk to them, as well as to local data tables.

Ideally, your suite should attach to SQL data anywhere, from any application, and should probably support ODBC.

Developer support

There's a big difference in the support required for a developer's toolkit and a user tool. The suites are struggling to make the leap. As a developer's toolkit, a suite must be supported by developer's forums on CompuServe, CD-ROMs full of sample applications and documentation, publications, and vast amounts of documentation. Some suite vendors are familiar with these issues and are investing heavily in developer support.

How they stack up

In the following sections, I assign a score from one to five for each of the categories I've defined.

Since this picture of suites is based on a developer's requirements, I'm pushing features like consistent look and feel, ease of switching between applications, and advanced features in the applications into the background, with interesting results. In the more "standard" comparisons, suites appear to be competitively balanced, with no suite having a clear edge. As programming tools, the view is strikingly different.

Lotus SmartSuite

From the developer's viewpoint, the Lotus SmartSuite doesn't score well. It has no common macro language, no open toolbox, and very limited developer support.

Lotus has clear long-term plans for a common macro language. LotusScript is already up and running inside Lotus' new application development tool, Notes ViP (reviewed in *Data Based Advisor*, November, 1994). However, the recently released version 3.0 of SmartSuite shipped without LotusScript in any of the suite's applications. Users of 3.0 must still contend with 1-2-3 macros, Ami Pro macros, and Approach, which in its current version includes no macro language. In terms of the application as a developer's toolbox, Lotus is committed to ripping open the applications so you can get at the internals, and has major effort in this direction under way. Like LotusScript, though, this work didn't make it into version 3.0.

Lotus SmartSuite
Lotus Development Corp.
55 Cambridge Parkway
Cambridge, MA 02142
800-426-7682, (617)577-8500
Fax (617)693-3512

Common Macro Language	●
Consistent Programming IDE	●
Open Toolbox	●
Data Links	● ● ●
OLE	● ● ●
Developer Support	● ●

Lotus Approach includes complete support for ODBC; Approach can work with any table via ODBC as if it had been created in Approach. Lotus 1-2-3, on the other hand, doesn't support ODBC, so a live ODBC connection into spreadsheets isn't possible. Ami Pro 3.0 doesn't support ODBC directly; the best that can be managed is the insertion of OLE objects containing the data, using Approach as the OLE server.

The major upgrade in SmartSuite 3.0 from the developer's perspective, is support for OLE 2.0. SmartSuite applications now support editing in place, but not OLE custom controls or OLE object programming.

Lotus has created a comprehensive developer support program, including a monthly CD for developers. Unfortunately, it's aimed at Notes developers, and nothing similar is available for SmartSuite developers—at least not yet.

WordPerfect's Perfect Office Suite

WordPerfect's Perfect Office suffers from multiple parents, and nowhere more so than in terms of programming support. The Office Suite requires developers to manage WordPerfect macros, Paradox PAL, and Quattro Pro macros. There's no sign of an open toolbox, under which developers could use the functions suite's in their custom applications. And there's not much in the way of developer support.

Perfect Office Suite
WordPerfect Direct Sales
MS 3234
500 South 500 West
Lindon, Utah 84042
800-451-5151, (801)225-8000
Fax(801)229-1566

Common Macro Language	●
Consistent Programming IDE	●
Open Toolbox	●
Data Links	● ● ● ●
OLE	● ●
Developer Support	●

Data links under WordPerfect are the best part of this story. WordPerfect can link and query a wide range of data sources: DataPerfect, Clipper, Paradox, and a variety of SQL-based servers. Paradox can use a variety of data sources through Borland's Database Engine.

WordPerfect's Office Suite supports OLE 1.0, but none of the advanced OLE 2.0 features.

Microsoft Office

Microsoft indicated as far back as 1987 that a common macro language was an objective—and they're still working on it. They still have a way to go, but the process is well under way. WordBasic and Visual Basic For Applications are very similar, but diverge on topics like function declaration and control statements. However, Access Basic and Visual Basic for Applications are close enough for an effective cut and paste in many instances.

Microsoft Office
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
800-677-7377, (206)882-8080
Fax (206)883-8101

A consistent programming IDE is also still in the future, particularly given the differences in the way programming is done in Access as compared to Excel. Curiously, WordBasic has the most intuitive development layout of any of the Microsoft Office applications and it's the one scheduled for the scrap heap.

The open toolbox concept was practically invented in Office. Microsoft calls it "exposed objects." Word, as well as Excel, show the results: The list of functions and objects available to developers is nothing short of marvelous.

Data retrieval in Microsoft Office benefits from ample ODBC support, but flaws still persist. Out of WordBasic, for example, ODBC is available only through an add-in that isn't officially supported by Microsoft. It allows direct SQL calls, but can only manage to return strings, so ODBC storage and retrieval of binary objects is out.

Of course, Microsoft includes the most advanced OLE implementation, and is the only suite that allows editing in place, and OLE object automation. It's also the only suite that supports OLE custom controls (.OCXs), the standard intended to supplant .VBX custom controls.

Developer support is impressive. Microsoft moved early to redefine Office as a developers' toolkit, and the release of the Microsoft Office Developers Kit CD-ROM, with its hundreds of megabytes of samples, SDKs, and documentation, is evidence of just how serious Microsoft is about supporting developers.

Common Macro Language	●●●
Consistent Programming IDE	●●
Open Toolbox	●●●●●
Data Links	●●●
OLE	●●●●●
Developer Support	●●●●●

An unequal struggle

From a developer's point of view, there's no contest; in their current versions, only one of the suites begins to fulfill the basic requirements of a development tool. Microsoft Office 4.2 is a powerful development environment, powerfully supported by Microsoft as a development tool. The others have interesting plans for the future, but little available for developers in their current incarnations.

Future considerations

None of the vendors or packages is standing pat; WordPerfect's Suite has by far the most ground to cover. The successor, PerfectOffice is in Novell's hands (they purchased both Quattro Pro and WordPerfect) and should be available. PerfectOffice, will include OLE support as well as a non-integrated common macro language, to be used as a layer on top of the applications. It will be interesting to see the results of this approach.

Lotus has been furiously expanding and upgrading SmartSuite. Lotus fans may complain that Lotus' relatively poor showing in this evaluation ignores the wonderful possibilities in Lotus Notes. The company recently announced a product called NotesSuite that combines the current version of SmartSuite, Lotus Notes 3.2, and a CD-ROM full of pre-built applications combining Notes with Smartsuite. A NotesSuite starter pack includes a Notes server (for NT, Netware, OS/2, or Unix) and two copies of SmartSuite 3.0 and Notes 3.2, all for just $995. Here, Lotus enjoys the same advantages Microsoft does with OLE—it owns the technology. That means Lotus will be the first out with the products that work together with Notes. For corporations committed to Lotus Notes, there are strong reasons for sticking with Lotus products. There's no doubt that SmartSuite combined with Notes, is far beyond SmartSuite alone as a development tool. ■

Article reprinted with permission from *Data Based Advisor*, Vol.13, No.1 © 1995 **Advisor Publications Inc.**, 4010 Morena Blvd., San Diego, CA 92117. For more company and subscription information, call (619)483-6400 or fax (619)483-9851.