For the last three years, we have tackled the topic of object-oriented data-
bases each September. For the first time, there's widespread commercial ac-
tivity to report, although customers still number in the low hundreds. 
Rather than product announcements, there are now product shipments. Rather 
than research studies, there are sales agreements, such as Objectivity's 
recently announced deal with Digital Equipment. And rather than benign ne-
glect, there are serious questions from customers and competitors alike.

Why object-oriented databases? Can't we just extend relational? Where are 
the standards? The ethnic controversy (as usual, a sign of underlying eco-
nomic tensions) between relational and object-oriented has heated up to the 
point that each side is now circulating a manifesto: "The Object-oriented 
Database System Manifesto" on the one hand, and the "Third-Generation Data 
Base System Manifesto" from "The Committee for Advanced DBMS Function." At 
least neither side calls itself "open." (See page 22 for pointers.)

Object-oriented databases are coming into their own, but there's still con-
fusion about what they are. There's a big difference between a relational 
database front-ended by object-oriented tools, applications or data models, 
and a true object-oriented database. In many cases -- for most traditional 
applications, in fact -- simply extending a relational database is the pref-
erable approach, especially since the data is probably already in a rela-
tional database and things work okay as they are.

Object-oriented databases are for data that is so complex and richly struc-
tured that it isn't in a database yet; generally, it's on paper, or at best in 
application-specific files inaccessible to other applications. OODBs are 
for new applications, such as CAD/CAM, computer-aided software engineering, 
heavy-duty text manipulation (not just storage and retrieval, but document 
generation and assembly, multi-page periodical and catalogue layout, version 
management and other more complex tasks), expert systems and other design 
applications. These are applications where you are manipulating not the 
data within the structures, but the structures and relationships themselves.

This is a key point, and it reflects the direction we are taking in the 90s: 
Routine tasks are being automated and are performed automatically. The area 
where we most need computer help is the design, integration, maintenance, enhan-
cement and modification of those auto-
mated routines. (For background on the 
vendors and on how object-oriented data-
bases work, see Release 1.0, 89-9, 88-
12, 88-9 and 87-8.) 

KILLER APP IS DEAD! -- PAGE 17

INSIDE
OBJECTS AT LARGE 1
Standards. 
Instances. 
OBJECTS ON THE HORIZON 17 
IBMetaphor. 
SYBASE OPEN SERVER 19 
Dynamic application exchange. 
PLATFORMS FOR COMPUTING FORUM 21 
Beyond the desktop... 
PHONES/RESOURCES/CALENDAR 22-27 
Sources, manifestos, dates.
The commercial news

The pioneers Servio and Ontologic, with 130 and 150 customers respectively, are launching second releases of GemStone, shipping for three years, and Ontos, the successor to Vbase and first shipped a year ago. Other firms who announced their existence a year or two ago are now announcing the existence of customers and the availability of products. They include Itasca, formerly Artemis, with Itasca, based on MCC's Orion; Objectivity, with Objectivity/DB and a cooperation agreement with DEC; Object Design, with ObjectStore; Versant Object Technology, formerly Object-Sciences, with its Versant line. Overall pricing levels are lower, down from as much as $25,000 for the first unit, to half that or less in most cases. Although a broad market is becoming visible, there seem to be almost as many vendors as customers.

"Everyone is still trying to figure out how to use OODBs," says Itasca sales & marketing vp Bill Hoffman, formerly with AI firms Gold Hill and Symbolics. "All the business plans add up to about five times the entire potential market."

Potential customers now have products to compare, reference accounts to call -- and the nagging suspicion that if they don't step up to object-oriented databases their competitors may get a jump on them. Pilot applications include not just electronics, but also office automation and software engineering. A variety of vendors of document and image-processing systems such as Kodak are testing object-oriented databases as foundations for their systems, while US West is considering OODBs not just for modeling telecommunications networks, but also for billing commercial customers where charges depend on complex interactions among assets and services, timing and amount of usage, regulatory constraints and other factors. All these activities serve to make the technology seem less exotic and more real and practical.

Customers are frequently buying evaluation copies from several vendors at a time. They include not just -onics and -tech companies and research outfits, but firms such as Martin Marietta, Boeing, General Dynamics, Mitsubishi, Citicorp, British Petroleum, IBM, DEC, Japan's NTT and most of the former Bell operating companies. An airline and a couple of hardware companies are looking at OODBs as a foundation for office automation. (No online transaction-processing, though.)

Standards: Object(ions to Object) SQL

The big sticking point is a lack of standards -- and customers' fears of being stuck with the wrong choice through lack of expertise and experience. Aside from words, you can't commit to object-oriented databases in general, and few people want to commit yet to one in particular -- which is why DEC's endorsement of Objectivity is so important. "Customers wanted this technology, but they weren't comfortable buying it from an independent third party unless it had real support from a system vendor [such as DEC]," says DEC's object-based systems manager Lydia Bennett. "Our commitment is real; we'll be making more announcements about Objectivity, including something with Trellis." Trellis is DEC's object-oriented development environment prototype -- highly successful and in evaluation at several large companies, says Bennett. (Only hardware vendors can afford such long beta cycles...)

Release 1.0

19 September 1990
the other hand, customers don’t want to buy a product that would run on DEC equipment alone, and Objectivity’s independence and multi-vendor stance is an important part of its appeal to its customers and therefore to DEC (cf. the relationships between IBM and Microsoft and Metaphor).

Isn’t defining a query/data manipulation language enough? Unfortunately, no. That gulf between programming languages and database languages is exactly what object-oriented databases are trying to bridge. It’s time to overcome the impedance mismatch between languages and databases, say the OODBers. The gulf between procedural and data languages was annoying; the gulf between object-oriented languages and database languages will be worse. The essence of object-orientation is the closeness of data and programs.

That’s the case made by Object Design, which has decided to standardize on C++ and offers an optional modification (implemented in its C++ Cfront preprocessor) to create persistent objects automatically and almost seamlessly. Competitors charge that this makes ObjectStore nonstandard, even though it’s optional, and ODI now stresses that you can also create persistent objects the "regular" way, with subroutines called from C++ or other languages. You still can’t use those subroutines for other OODBs, nor can you use theirs for ObjectStore, so this strikes us as an overblown issue for ODI (but one ultimately needing a standard solution for everyone). Meanwhile, some customers who want an OODB rather than an illusory safe solution appreciate ObjectStore’s seamlessness.

All for one or one for all

On the other hand, the ability to segregate all the database issues in a separate language is labor-saving in the extreme: Who wants to extend all current computer languages, especially if we want them all to be able to interact with the same database? The extensions would have to be similar, so they would fit awkwardly with each language anyway. Those who most want to put the database into the language are generally working on the assumption that there’s only one language that really matters -- C++.

Servio’s Jacob Stein: "We OODB vendors need a dairy council: ‘Drink milk; it’s good for you.’ We don’t even care what brand."

As people outside the C++ world come to see the value of OODBs, we expect the world to resign itself to the gulf between programming and database languages. But it would be a lot easier if OODB vendors could at least agree on an Object SQL standard, which would also provide a bridge from the relational world. Work on such a standard is going on at several standards groups, including the CAD Framework Initiative and the ANSI OODB Task Group (page 16). These standards will specify how to tell any database to store an object, and shouldn’t require much work for all the vendors to support as subroutine libraries. Unfortunately, they won’t quickly solve the really tough problems of interoperability and interchangeability of objects at a meaningful level. The goal is to standardize not just on how to get objects in and out, but how to determine their structures and relationships, how to manipulate them, make them persistent and the like. How do you override inheritance; how do you indicate no value or no information versus 0 or "definitely not"? Precisely because objects have more character than simple data elements, there are more issues to consider.
"New, improved, now with objects" vs. "Object-oriented from scratch"

There is a fundamental difference between databases with objects and object-oriented databases. The first extends the relational database model by allowing it to include objects where you would expect a simple string or value in a field, and perhaps by adding some interesting syntax to the data language to handle hierarchies of inheritance or composition (is-a-kind-of and is-a-part-of, respectively), and relationships such as distance. The other is object-oriented all the way, with a storage system that maintains the relationships among objects and between objects and methods, and that clusters hierarchies together and (ideally) locks objects, not pages.

Some argue that you can extend the relational model enough to make it a full-fledged object-oriented database. Oracle senior architect David Beech, for one, posits that relational is just a special case of object-oriented: A table is a special kind of class; the rows are objects, with relationships listed in the fields as keys instead of pointers. A part, for example, could have a list of all the items that use it. All we have to do is handle more kinds of objects, while keeping referential integrity, portability, reliability, SQL and the other wonderful things relational brought us. This would give you a structurally object-oriented database, but the structure would be added on, with relationships listed in tuples that are derived by value matches rather than directly -- a process that gets slow in extremis.\(^1\)

History teaches that pioneers start fresh; they may start from others' shoulders, but they build their new implementations from scratch.

Another important component of an object-oriented database, the methods and their binding to objects in multiple hierarchies, can be represented as applications that reside in the database at the server rather than out there with the client applications. Sybase with its stored procedures and Ingres with triggers and some other extensions have already taken serious steps in this direction. If the database supports methods (encapsulated procedures, or executable code) and the process of inheritance, overriding and other object-oriented processes, it is behaviorally object-oriented. But again, this is a hacked-on kludge instead of a system intimately supported and maintained by the database. (You can also do object-oriented programming in C or even COBOL.) The point of an object-oriented database is that the database knows about these things and makes them work consistently.

The function of the object-oriented database is not just data storage -- which it may do with its own optimized, physical-level capabilities or through the medium of an RDB as makes sense -- but maintenance of data integrity at a higher level than an RDB can. That is, a relational data base maintains the integrity of the data elements, while an OODB manages the consistency of the object structures -- often over long periods of time where

\(^1\) There's no reason that relational vendors can't move to other physical storage systems, says Beech, with object-oriented clustering, access methods and indexes. (Of course, you can also denormalize, but if you do so your database is no longer truly relational...)
For example: Fold me in your database of applicants

How would an OODB work as a foundation for office automation? There are lots of answers depending on the specific tasks involved, so let's consider one example: a job-applicant-processing system. The basic object is simply the folder -- just one step away from the file metaphor. A folder typically includes a certain amount of table-style data: name & address, phone numbers, dates, qualifications, previous jobs, previous employers, etc. But the folder may also include a wide variety of supporting items -- the text of the resume, an image of the resume and perhaps the candidate's photo or a news clipping, notes from interviews with the candidate or references, supporting letters, phone-call memos, voice notes, test results, and appointment times and deadlines related to the selection/hiring process. Some of this information may "belong" to a particular recruiter working with the candidate, but other workers in the group may need or want access to some of it. The folder or selected contents can be sent around the office or outside it; its route might be preset and monitored, or the activity could be more ad hoc. (We will discuss these ideas at length next month in our issue on groupware development tools.) Items within the folder could also have behavior, or methods: How to display themselves, how and where to send themselves, when to send a follow-up request for information, and so forth.

For example, when Juan hires Alice, the accept-candidate method triggers a number of actions: A letter is sent to Alice, with information such as Alice's name and address and start date selected from the folder. Other information comes from other sources: The letter itself comprises text from a sub-database of document objects -- paragraphs about the department Alice is joining, about the product she'll be selling and about the glorious history of the company itself. The job description comes from a listing of job requisitions -- another kind of folder, with a collection of jobs that contain both structured data such as salaries and departments and job titles, and less structured information such as requirements and duties.

A notification is also sent to the employee newsletter, including a scanned image of Alice's photograph, and to the personnel department if Juan made the decision all by himself, or to Juan if the application is run by the personnel department. Another message might be sent to the payroll department, either to a person or direct to the payroll application, with proper authorization. The company phone directory (kept online, of course) would also be updated. In a really automated company, another message might generate purchase orders for new computer equipment and software for Alice unless she is replacing someone, and create a new node for her on the department LAN.

To each part of the system, Alice-as-object has different characteristics: To one, she is the recipient of a letter, to another she is a computer-user, to yet another she is an employee record. The OODB maintains the "Alice object" and delivers it and related objects properly for each user/application. While it appears as an object within the object-oriented database and the object-oriented applications built around it, it may also be a record in a number of relational databases updated by the OODB. It's a heterogeneous world.
the conflicts must be recognized and maintained as unresolved but needing reconciliation ultimately. Trying to represent these constraints as applications is complex, and subject to error and bypass by other applications. The whole point is to get them into the database -- for use by applications rather than as peers to applications.

---------------
Object-oriented databases represent the ultimate in reuse and sharing of code. The complex interrelationships and interactions of objects and methods need to be represented only once...and are then maintained for all users and applications. Application programmers avoid redundant efforts.

---------------

Hybrid vigor? Or just a mediocre jack of all trades?

Beech is right, in theory. At some point, some of these extended relational databases may work as well, but it won't be that one side or the other has won. Rather, the two will merge and hardware will handle performance.

The question is whether people who grew up in the narrow world of relational data and files have the mindset to make it in the broader realm of objects. Much as we may think we're dealing only with technology, and high technology at that, we're also dealing with subtleties and attitudes -- the same things that keep most mainframe people from designing good interfaces, most C programmers from learning Smalltalk, and most reporters from writing poetry.

In the end, relational databases aren't optimized for the tasks OODBs eat for breakfast. DEC's Lydia Bennett, who spent more than a year making business and technical evaluations of the leading contenders, ultimately concluded that extended relational systems just couldn't cut it on the tasks for which OODBs are suited, although in other cases, with mostly record-oriented data with objects attached, they work fine. It boils down to the relative importance of structure versus data values.

Of course, customers don't like change any more than vendors do. They want their new problems solved in new ways, but they don't want much disruption, just incremental improvements, in the systems they've built with great pain and expense to handle their existing problems. If it were a question of moving from RDB to OODB for the same applications and data, the evolutionary approach would likely win, but this is new stuff for new applications.

Peaceful coexistence with territorial integrity

The traditional second-generation database vendors are amassing their forces to stop the onslaught of the object-oriented database vendors -- unnecessarily, we believe. Co-existence is necessary. Relational databases cannot hope to do all the work of the object-oriented databases, nor can object-oriented databases hope to hold all the data and handle all the routine transactions that the RDBs currently do. This war may be something of a bust for the warring factions (albeit a good source of military spending for the marketers and other support troops).

The essence of object-orientation is run-time binding and ad-hocness of procedures; if you're following regular procedures, you can get better perfor-
mance with canned, procedural applications; and if you’re performing regular procedures with regular data, you can get the best performance of all with an old-fashioned Codasyl or network database. It’s important that databases work together, not that they take on each other’s functions or capabilities. In many cases, object-oriented databases will call on RDBs for data. You could even get at an ISAM database using an object-oriented front-end (see page 19 on Sybase’s support for access to mainframe database applications).

DEC has helped the case both for OODBs and for coexistence, by stating that Objectivity is "complementary" to its own relational database, RDB, and the de facto standard on DEC and UNIX, Oracle. "We have the ability to be impartial as a database vendor and sell both kinds," says DEC’s Bennett, "and to understand that we aren’t cannibalizing our relational business." The real de facto standard is SQL -- all the other differences, you might argue, are just implementation details. The implications of this are interesting. First of all, it seems that DEC has decided that it’s important to have an object-oriented database, but it realized from Oracle’s success that customers don’t want a "proprietary" system -- i.e., one available only from DEC. (In a hardware view of the world, software from a single vendor available on a variety of hardware platforms evidently doesn’t count as proprietary.)

---

From the point of view of the RDB, the OODB is just an application.
From the point of view of the OODB, the RDB is just a source of data.
---

Right now, the world is caught up in an insistence on standards and finding the one right way to do everything. Longer-term, we may realize that there are many ways to do something, or more accurately, we may better understand the differences among the things we’re trying to do. (In the same way, you need different tools for handling data on a desktop from those you need on a notepad; this makes sense once you recognize that the tasks also differ subtly from platform to platform. Data transfer, yes; slavish imitation, no. See Release 1.0, 90-6.)

The result: Warlike coexistence

Will the relational vendors suffer the fate of Cullinet, which ignored relational until it was too late? In the end Cullinet’s problem wasn’t that everyone switched to relational but that new applications went to relational while most old ones stayed with IBM. Moreover, the data the relational systems hold is fundamentally the same data the old Codasyl and network databases held, making conversion difficult but a reasonable thing to do. (Even so, not all that much has been converted; it’s mostly new applications with new data or decision-support with the same data).

By contrast, object-oriented vendors and users are generally dealing with new data. Thus traditional relational database vendors needn’t fear that

---

2 They might argue that all data is object-oriented; we’ve just limited its richness artificially in order to stuff it into nasty, dry, restrictive relational databases. Yes, but when you’re doing online transaction processing you really don’t care about holistic data; you care about speed and one or two data elements and that’s all.

Release 1.0

19 September 1990
users will switch to object-oriented databases. What they should fear is that the focus of greatest growth has moved on, from automating data-intensive tasks to assisting in the performance of structure-intensive tasks such as design, configuration and the maintenance of knowledge bases.

Common sense

While building an object-oriented database is a complex task, it's not the mystery it may seem. Like artificial intelligence, it leaves you asking, Is that all there is? Like AI, it's not impressive on a small scale; its magic is managing a huge volume of things consistently rather than handling any single small thing with insight or true "intelligence," whatever that is.

An RDB gives you one kind of flexibility: Each data item appears once and can be linked at runtime in any way based on matching values. An object-oriented database gives you a different kind of flexibility in application programming: Much of the application can be written at a high level, and the actual implementation is determined at runtime based on the objects used.

The OODB vendors are wrestling with the tough problems of optimizing data storage and managing high-level consistency across users, databases, transactions and time. Consider an object-oriented database as one where the indexes are more important than the data. The relationships are maintained to other objects and to each object's own and inherited attributes and behavior. That is, the system maintains the relationships between the items with permanent pointers, and optimizes the storage locations to reflect those relationships; the values within the items are secondary. By contrast, a traditional database simply holds the data in sequenced files and lets the application run through them; a relational database lets you get at the data by values in fields, including joins where values match. The RDB is clean and abstract, instead of preconfigured, and it means that there's a lot of redundant application code, even though there's no redundant data.

Who's using them? What for?

The first, leading-edge users of object-oriented databases tend to be engineers or designers. They routinely work with enormous numbers of elements which interact with each other, fit together, and share features and behavior -- call them instances of objects and classes. The users of the OODBs work together on large, complex, interacting systems; they change things in processes that may take minutes or days; they use a variety of applications to handle different aspects of the design process. They also want to reuse their work and share it with others, possibly customers or subcontractors.

Companies such as McDonnell Douglas, Boeing and the various chip-makers are on virtually all the companies' customer lists, as are vendors of the software tools they use. Few design-tool vendors have yet standardized on a single OODB, but exceptions include Valid for Objectivity/DB, Associative Design Technology for Ontos and InterACT/Advanced Computer Techniques for GemStone/Servio. Index, as noted below, is using Ontos for its forthcoming repository. Fluent Machines, founded by Apollo founder Dave Nelson, is also using Ontos in its new OS/2-based multi-media development and delivery plat-
form. Dave Parkhill, who leads a research team at US West, says he's using object-oriented software and ObjectStore in order to reduce prototyping costs and shorten time to market, but it may also be the foundation of production systems for commercial billing systems and network management as well as design. Many customers are still in the exploration or the not-ready-to-announce stage; Servio is working closely with a financial institution and an airline, but can't persuade them to reveal their identities.

In a typical design scenario where an object-oriented database could help, transistors are connected to circuits with specified behaviors and electrical properties which must correspond. Assemblies of these elements are combined to form larger components, and so on until you have an entire chip with hundreds of thousands or millions of discrete parts. These chips then work together to form an entire subassembly, which will ultimately connect to others to make a complete system.

--------

Index: The repository knows

Index Technology is in the throes of building the next generation of Excelerator CASE tool on top of Ontos. It will manage all the information -- specifications, diagrams, source and object code, documentation, and all the versions thereof -- that developers work with. (The executable code modules themselves need not be object-oriented and will generally be stored in files resident in and compiled for a particular environment.) The Index project is progressing somewhat more slowly than expected, but more because of the CASE and OS/2 aspects than because of Ontos. An OS/2, c-tree version will be out in a few months, says Index's vp software development Jerry Katzke, but the version built on Ontos will take another year or so. Why wait? "Ontos gives us a more robust data model; now we have to build the data model on top [rather than let the database manage it]. And with Ontos we can put rules in around the data as part of the system, rather than as external constraints you can bypass. Finally, Ontos will make it a lot easier to support versioning and configuration management and workgroup computing." (All these benefits would be true of any object-oriented database system.)

CASE repositories such as Index's aren't just for development projects; they're where you manage reference copies of software permanently. Copies can be used for execution, but this is where you go for modules when you want to do something new; to maintain integrity when you change something; to find a data definition to reuse; to coordinate when you're building a system to interact with another department. The repository knows who is using what versions of what software. And, because it's an OODB (not a real-time relational db with instantaneous two-phase commit), it can be made to handle the ambiguities of real life: Not all the information is on-line; some units get out of touch and must be re-synchronized on a periodic basis. The OODB knows what has changed and what hasn't, where reconciliation can be automatic and where a human has to come in to resolve conflicts. (Note that the OODB itself doesn't resolve conflicts; it just keeps track of them, and rules or embedded processes in the database take care of the semantics.)

In a better publicized project, IBM and its cohorts are working on the IBM Repository, a DB2-based database extended and front-ended to handle objects which will compete or interact with truly object-oriented systems from such
vendors as Index Technology. We suspect that at some point it will become clear that OODB would have been the way to go.

Beyond WKS and WK1

Lotus is looking at a number of object-oriented databases in a project managed by Toby Bloom (who formerly worked on the Index project) under the guidance of chief scientist David Reed. The goal is not a new foundation for Notes, which already has its own rich database, nor any change in direction of the company's relational database efforts, but rather a way to manage the complex data in spreadsheets and graphics now stored as files. Files don't provide the necessary granularity or representation of dependencies. For example, you may want to maintain multiple versions of a spreadsheet where only a few cells and formulas differ from version to version, or consolidations that consist of a number of linked spreadsheets with different data for different regions or products (cf. Lotus's Improv for NeXT). In other words, the objects could be cells or collections of cells (cf. named ranges) or formulas or macros, not whole files as in, say, NewWave.

"We want all of the benefits of database -- reliability, sharing, integrity, transaction management -- without the abstractness of SQL," says Bloom. "We want to be able to associate procedures [formulas] with small-grained items of information. And we want to be able to share them across different versions of our own [application] software." (For an example of an environment with similar ideas, see the discussion of Patriot Partners, page 17.)

-------------

For now, object-oriented databases are hard to understand, hard to benchmark (because performance and needs are so application-specific) and hard...ly finished. Buyers will be limited to astute people who understand both their own problems and how OODBs could solve them, and who feel comfortable with the technology.

-------------

Updated data on instances of the class OODB

To use objectspeak, the general class of object-oriented databases is beginning to specialize itself into subclasses which inherit some common characteristics but also differ in a variety of ways. It's difficult to compare and contrast these systems, since their vendors generally consider them optimized for all tasks except possibly OLTP. (We described them all from a technical perspective a year ago; see Release 1.0, 89-9.) Most are now shipping, but not all have the courage to call their products finished; Versant and Object Design are selling "revenue-beta" versions.

-------------

3 However, it's not object-oriented, because it doesn't (easily) allow for the creation of new classes, nor does it support object hierarchies in general. However, you could easily put its information into an object-oriented database. That wouldn't give you automatic interoperability between Notes and, say, most wp or data formats, but it would make it easier to handle the process of conversion automatically once you had written the conversion routines. In other words, it could make managing the integration of Notes with other systems easier for programmers, and virtually invisible to end-users.
The basic difference seems to be between the Smalltalk/LISP/runtime-binding/interpreted camp and the C++/UNIX camp. These are broad distinctions and generalizations, but they divide the world into Servio, Statics and Itasca on the one hand, and the O brothers -- Object Design, Objectivity, Ontologic and Altair (with O2) -- on the other. Versant, a special case with a strong relational flavor, and Servio, with its new C++ interface, put themselves in both camps.

The first group are truly object-oriented, they will tell you, and store their methods right with their objects. The methods get interpreted at each client where they are executed, making for maximum portability (if not performance). Except for Versant, which has a heavy relational culture, these vendors sell to people who may be less concerned with performance and runtime execution than with ease of prototyping and flexibility.

The C++ crowd generally store their heavy-duty code on a particular target machine/environment, where it is compiled for that environment and can run faster. That means faster performance in execution, but slower performance in modifications (because the changes need to be compiled).

The distinction is much like that between interpreted languages and C/C++ in general. Again the question is, Are you spending more time designing the components, or managing the assembly and interaction of components someone has already designed? That's a fine distinction, and partly just a question of levels of abstraction and granularity. Someone designing circuit diagrams or mechanical systems is probably manipulating objects or putting data into them without modifying the code of individual ones as much, whereas someone using an object-oriented system to design a prototype is probably writing and adding a lot of new code/objects. The first person would go with C++, while the second might go with Smalltalk. Likewise, office automation or publishing applications could go either way, depending on whether you're producing multiple documents or designing new ones -- publishing periodicals with standard formats and varying content, vs. designing new documentation systems or changing the layout rules for each customer in a job-shop printing house.

In the end, customers have to decide what's important to them and make their own choices. Within either category, there are finer issues, such as how much flexibility you get vs. how much designers or programmers need to know about system innards, speed of access vs. speed of updating (cf. query vs. transaction performance in a relational database).

Object Design -- A kinder, gentler Oracle

On the East Coast, Object Design is winning new business and quickly turning into the Oracle of the group, with aggressive marketing and two salespeople (of four) from Oracle, led by ceo Ken Marshall, who ran Oracle East Coast operations. "We want to be a kinder, gentler Oracle," says founder and chairman Tom Atwood, who earlier directed much of Ontologic's effort. Object Design seems to be catching up fast in visibility if not in numbers to Ontologic, which has an early lead and a 150-strong customer base.

Object Design now has about 20 customers, all in beta, for about 100 individual users overall. They include installations at Microsoft, Lotus (page 10), NEC and Olivetti. US West and Martin Marietta are using ODI's Object-
Store for several projects, including network management (more a logistics and configuration management application than real-time switching). Object-Store costs $9000 or less per seat, and is scheduled commercial release next month. Already, customers’ enthusiasm, among other things, has helped the company win a second round of financing from Vista, Harvard Ventures and NEA, as well as Kodak, Olivetti, Philips and Orien (representing several Japanese manufacturers), for $9 million in total.

The company’s ObjectStore gets good marks from Lotus’s Bloom and Southwestern Bell’s Todd Bashuk among others. It has a virtual-memory analogue for data that traps hardware faults generated by accesses to stored objects (rather than using software to check whether an object is persistent, as most of the others do). This works fine as long as you’re on the Sun, the current platform, but hasn’t been shipped for other systems yet. Performance is superior as long as most of the objects are in memory (as with virtual memory for code), but it suffers when a lot of disk swapping is required -- as it does with any database, object-oriented or not.

So far, customers don’t seem to find that a problem, even with more than 300,000 (small) objects or, says Kodak’s Bob Gordon, 50-megabyte objects (images). However, ObjectStore provides limited flexibility in options for concurrency and short transaction control, with few options for granularity of locking or timing, but it has extensive support for long transactions and version control.

For better or worse, ObjectStore offers the most seamless environment, with tight links between the database and its version of C++. Although Object Design supports the usual standards -- UNIX, C++, Motif -- and graphical tools, the company sees itself as a leader in object-oriented technology itself; its optional seamless C++ is either proprietary or uniquely enhanced, depending on your point of view.

ODI also has the distinction of having licensed its parameterized type technology to AT&T for use in a future release of C++. Parameterized types bring more object-orientation to C++ than it had before, make development faster and easier, and fulfill a specification of the ANSI C++ standard. In brief, you can create a class such as a generic set and let the system deal with it as a set, and only later create a specific instance that holds boats, and then it binds in the boat-specific methods.

Ontologic -- Mature and full-featured

By comparison, Ontologic’s Ontos allows for great flexibility in locking by objects and transactions...but a lot more complexity. The choices you get to make, you gotta make. Overall, Ontos is the most mature of the C++ products; it has more tools than any system except GemStone, and it is the only one yet to run on OS/2. Along with the second release of Ontos, Ontologic recently announced Shorthand, an object-oriented 4GL (easy-to-use application language) for non-C++ users (which generates C++, obviating the non-portability arguments against most 4GLs), and a graphical screen generator. Whereas Objectivity sells mostly to OEMs and true techies, Ontologic and Servio seem best-equipped to sell direct to users (though even these are programmers rather than your typical power PC user).

Ontologic’s customers include General Dynamics, British Telecom, NTT, British Aerospace, Motorola (for a manufacturing application) and Xerox (a re-
search group in Rochester). On the strength of its recent success in garnering customers, the company recently raised another $4.3 million from its investors, which include Kleiner Perkins, Wolfenson, Smith Barney and Phoenix, as well as Patricof, a new investor.

Objectivity -- Object macho

Objectivity has garnered attention for its deal with DEC, as well as agreements with Sony and Valid Logic Systems. Shipments began last April. Development systems cost $30,000; runtime systems, quantity discounts and OEM licenses are available. Objectivity/DB runs on Sun 3s and 4s and DECstations under UNIX, with VMS and VAXstations on the way and others to follow. The system is designed to be fully distributed, with databases and clients anywhere on a multi-vendor network. It has a multi-level cache, and user-settable options for granularity of concurrency and transaction control.

Funded by Mayfield, Institutional Venture and Menlo Ventures, Objectivity is focusing on the engineering and scientific market, says president Bob Field. He is a former venture capitalist and sales manager at four successful start-ups; most of the other employees come from electronics or computer firms. Objectivity/DB comes with application notes and sample structures and code for an EDIF (Electronic Design Interchange Format) circuit model and netlister, a PDES (Product Data Exchange Specification) geometrical and topological 3D model, and a software configuration manager. The deal with DEC will help it reach other markets such as office automation, manufacturing and finance without diverting its own resources and attention.

More than anything, the company is focused on "a back-end database, for full size, full performance and full reliability," says co-founder Drew Wade, formerly with Daisy and HP.

Altair -- Objets de France

Paris-based Altair is a government-assisted consortium (50 percent) with investments from Siemens (34), Bull (10) and a university -- $25 million altogether). It is coming along on schedule with 02, a C- and UNIX-oriented object-oriented database. It manages storage by pages, reducing granularity but increasing performance from the earlier object-based scheme. The project will go commercial and independent next year, with minority shares from Bull and Siemens and a large part of ownership in the hands of founder Francois Bancilhon and other employees. Altair plans a beta release late this year to ten customers, nine in France and one in the US. The system will

--------

4 Valid will use Objectivity/DB as the foundation of its Design Manager, a networked back-end both for Valid's own suite of electronics design tools and for others its customers use, says senior product marketing manager Larry Rice. Like many companies in this consolidating business, Valid has collected several different tools through acquisitions, and needs a way to get them to work together. However, Design Manager initially will be more a way to manage users' work in the large than to manipulate the contents of the work. That is, it will manage a user's work, and the relationship, say, between a spec and the implementation, rather than the individual components he's designing. It should be available early next year.
start at about $30,000 per seat, with discounts for multiple copies. The
target market is mostly technical applications, but, says Bancilhon, "We
spent at least a third of our effort on Motif-based programming tools with
graphical, active, editable objects on the screen."

Servio -- Joining the mainstream

Servio's GemStone has been shipping the longest of all these systems, since
January 1988, and is now in its full-fledged release 2.0, including a new
C++ interface, hooks to Sybase's SQL Server, and support for a broad range
of platforms working together simultaneously including workstations and
servers from IBM, DEC and Sun, and Mac and PC clients. It is written in C,
and will shortly have a range of graphical tools in addition to Release
2.0's graphical schema designer.

The company has 130 customers, including IBM, Nippon Steel, US West, Martin
Marietta, JPL, Lockheed, Kodak and British Petroleum, for applications rang-
ing from CAD/CAM to medical information systems, office automation and cart-
ography. While performance has increased four times, the company says, the
pricing has dropped. It starts at $20,000 for a four-user license. Over-
all, Servio approaches Ontologic for the largest installed base. The compa-
ny has a strong offering, and the current version, with support not just for
C, C++ and Smalltalk but for applications in COBOL, FORTRAN, Ada and LISP
through RPCs, overcomes the company's traditional Smalltalk flavor which put
off a number of commercial accounts. Not only does it support both, but you
can access the same database from Smalltalk and C++. Over time, the company
has dramatically improved performance and debugged the system, to the point
that it's now robust enough to be used in production applications at Evans &
Sutherland, which is using it for daily, live management of parts libraries.

Aside from technical issues, the company, privately owned by an Indonesian
corporation, has recently made a number of moves towards the everyday world,
including the installation a year ago of president and ceo Donal O'Shea, who
was previously president of UniSoft and vp operations of the Open Software
Foundation. New marketing vp Carole Rutter worked at IBM for 24 years; Paul
Butterworth, chief scientist at Ingres, is joining Servio this month.

Itasca -- Orion reborn

Itasca, renamed from Artemis and derived from MCC's Orion, has been shipping
beta versions of its Itasca OODB since January, and production versions
since July. The system is optimized for multi-user, distributed use, with
the schema replicated at every workstation, data stored across the system in
locations optimized by the system (by default; the designer can also specify
where things should go), and a combination of private and shared databases
to minimize concurrency problems. It runs on Apollo, HP, Silicon Graphics
or Sun workstations, but not yet on mixed networks. It supports a C inter-ace plus its own database language, and will have a Motif-based GUI soon.
It has pessimistic locking at the object level (vs. pages for most competi-
tors who focus on performance), which maximizes integrity and minimizes un-
necessary conflicts because of the fine granularity of object-locking.
Overall, Itasca is elegant at handling complex object relationships and prototyping. Customers who have been using it are enthusiastic, but a lot of people dismiss it because of its provenance in the LISP world of MCC -- the very background that gives it its elegance and flexibility. Itasca's 15 customers so far are mostly government outfits or internal projects at commercial companies, including Allied Signal (3D models), automakers and aerospace companies. "We have competed in almost every account," says Itasca vp Bill Hoffman. "For now, we're piggy-backing, since the incremental cost to try us too is so low" at $3995 per server. Says one customer, proving the point, "For $2000 it's worth looking at if only for the intellectual experience." (This customer gets a discount as an MCC member.)

Symbolics -- Statice quo

Statice is one of Symbolics' stronger products right now, but the company lacks the resources to market it as aggressively as it deserves. Still, installations are up to 85 from 70 last year. Symbolics plans to use Statice as the basis of an interactive scheduling/planning business modeled on the successful Houston Power & Light installation (see Release 1.0, 89-9).

Versant -- Relational heritage, object-oriented nurture

Versant Object Technology, formerly Object-Sciences, is the least visible of the recent crop of start-ups. It has been shipping Versant ODBMS and related tools since June; the product costs about $15,000 on a variety of Sun workstations and the IBM RS/6000. The system is highly modular, with an underlying object-oriented data storage component, Versant ODBMS, and a logical layer, Versant Star, that implements the database capabilities -- transactions, versioning, schema representation, distributed data access & query and updates (but not to RDBs and OODBs in a single transaction). Versant Star acts as the insulating data schema manager between any kind of application system/language on the top and any kind of storage manager on the bottom (although it's assumed to be Versant ODBMS if it's object-oriented). "Post-beta" deliveries will start in early 1991, with support for Smalltalk-80 and Oracle and the Versant ODBMS through Versant Star.

President Mike Seashols says the company is ahead of plan, with $1.5 million in revenues so far, and has done better than expected in the MIS/OA world. At OOPSLA next month it will announce agreements with both a hardware vendor (as a reseller) and a large MIS customer (one of the top 20), he adds. This is a good sign for OODBs in general, since most people figured MIS interest wouldn't kick in until later. Certainly Versant is the best equipped to sell into this market in philosophy and in personnel, many of whom have backgrounds with Oracle and Ingres. Funding of more than $8 million comes from TA Associates, Bessemer Venture, Advent and Singapore Technology.

Overall, the company is a big believer in coexistence. Says Seashols, who ran sales and marketing for both Oracle and Ingres, "I was a relational guy and now I want to play in the object world and keep the benefits I got accustomed to with relational. We've gone a little bit higher in the food chain [than competitors], beyond design to production as a whole. It's our background; we came from selling to vps of MIS. I'd always been concerned about how fast that market would mature, but now it's coming along faster than I hoped. Manufacturing is about half the world, so we're looking at concurrent engineering, integrated manufacturing. The federal government's involvement in CALS is rippling through the manufacturing community."
And on the defending side...

Oracle is still assessing the issues, says Beech. The company hasn't announced anything yet, but we expect it will in due course. "It's not really the role of standards groups to implement things," says Beech, "so it seems to be the socially responsible thing [for commercial vendors] to participate in this as soon as you have something sound rather than go too far down the road. This is a real chance for convergence. We're ready to move ahead now and put in a little richness while still keeping control."

Ingres has been taking the lead in the offensive defense, although things may change with its acquisition by ASK and indirectly HP. The company was co-founded ten years ago by Berkeley professor Michael Stonebraker. Stonebraker is a co-author of "The Third-Generation Database Manifesto," along with Oracle's David Beech (who formerly worked on HP's Iris), in which they argue for saving the best of relational databases and extending them with support for objects. Ingres 6.3 (the current version) can handle objects and allows user-defined functions, but it's a long way from being object-oriented. "We'll become objectized before OODBs become productionized," asserts Ingres product manager David Kellogg.

Sybase, like Versant, offers an interface to almost anything with its Open Client/Server architecture (page 19), and could easily front-end an OODB. But unlike Versant, Sybase comes to the party with an established RDB business (although it lacks an object-oriented storage manager).

Hewlett-Packard, which now owns part of Ingres/ASK, is also making plans for 1991 to deliver a commercial implementation of Iris, its object-oriented database project. The system uses the same underlying storage manager as HP's Allbase/SQL relational database and will be extended relational with an SQL interface and support for NewWave, says project manager David Wertheim.

Open Objects: The dairy councils

There are three major standards groups with active OODB efforts: the Object Management Group, the CAD Framework Initiative, and ANSI, with task groups on C++ and OODBs. In short, all the companies are active in most of these efforts. The details: Object Design, Objectivity, Ontologic, Servio and Versant are all members of the Object Management Group. (All except Ontologic are also members of the Open Software Foundation.) Servio's Carole Rutter is on the board of OMG, and its David Taylor and Jacob Stein are active in subcommittees; Object Design is hosting the next OMG meeting devoted to OODBs. The CAD Framework Initiative is a group of vendors and users devoted to interoperability among design-automation tools; members active on its Storage Management subcommittee include Objectivity co-founder and technical wizard Drew Wade, who is chairman, as well as Versant, Ontologic and Object Design. Its other members include AT&T, Cadence, Computervision, DEC, GM, H-P, IBM, Mentor Graphics, Sony, Sun, Texas Instruments, and Valid. Ontologic's Tim Andrews chairs the ANSI X3 committee on OODBs, where Object Design, Servio, Versant and Objectivity are also active, and Object Design is particularly involved in ANSI's C++ work, including ODI's implementation of its draft standard for parameterized types. Itasca is currently using MCC's membership in OMG and CFI, but will join on its own shortly.

Release 1.0

19 September 1990
IBMETAPHOR: OFFICE OBJECTS WORTH WAITING PHOR?

The killer app is dead! Long live the killer lib! Users want not a single, awesome application, but a suite of intelligently communicating applications. To build them, developers want a platform-independent environment with built-in components and some standards for interoperability -- along with confidence that the platform will be commercially successful.

One of the most pernicious problems application developers face is the choice of an operating system/environment to write for: DOS, Windows, OS/2-PM, UNIX with Open Look or Motif or whatever. Database application developers can pick a database tool and (with a little tweaking) build something to run on the variety of platforms the database vendor supports, but office automation vendors aren't so lucky. They have to deal with a variety of word-processing formats, spreadsheets, interfaces, calendar formats, and the like, many of them from other vendors. This, of course, is precisely the kind of rich, structured, complex data described above.

In that context, the recent announcement of Partiot Partners, a partnership between IBM and Metaphor Computer Systems (in which IBM already has a $10-million investment; see Release 1.0, 86-13, 88-4) is a promising one -- if only the results weren't so far off. The frequently touted advantage of object-orientation is reusability, but this promise is conditional: All code is reusable, but it must also be worth reusing, and modular enough so that you can reuse or extend discrete parts of it in new combinations. Object-orientation makes these benefits easier to achieve, but it does not by itself guarantee them.

The background

A variety of companies have built their own object-oriented applications, but they tend not to encourage third-party participation. On the other hand, a number of vendors have built foundations for such systems, but without much embedded application knowledge. Hewlett-Packard's NewWave, for example, has the framework, but little to put inside it. There are a few applications been converted to be NewWave-aware such as 1-2-3, Excel and WordPerfect, so that they can work as the methods of NewWave objects such as a spreadsheet or wp file, or a chart. But there are few NewWave objects at the level where an application might manipulate objects, instead of just being an object, a monolithic set of methods attached to formatted data files. (That is, a WK1 file plus 1-2-3 is an object, but 1-2-3 itself does not know about objects; its cells are internal data structures.) In short, NewWave is an object-oriented environment, but it has very few classes of objects and relies on DDE for "object" exchange. Also, it runs on top of DOS (with Windows), with UNIX and OS/2 on the way, but it doesn't insulate developers from the underlying operating system or architecture.

By contrast, the Patriot system should offer a rich array of objects for a variety of applications to manipulate, starting with text and data objects, voice and image. To illustrate, the text objects would not be just generic "text," but subclasses such as headlines, paragraphs, footnotes, which a developer could use to construct indexing or layout functions for a word-processor, or to add limited text-handling to a spreadsheet.

Like Interleaf with its Active Documents technology (see Release 1.0, 90-3), Metaphor starts off with an application, its Data Interpretation System --
or at least a set of specific capabilities rather than an abstract notion of an object-oriented environment. The point of the DIS system is to do something in particular; later on you can marvel at how easy it is and wonder about generalizing the technology and extending it to other tasks. From a marketing point of view, the customer buys it as a solution to his problem, making sense of marketing data -- not to the developer's problem, fast design and implementation of easy-to-use applications.

The precise product Patriot Partners will introduce, or when, is unclear at this point, not just to us but even to IBM and Metaphor. The goal is to offer an object-oriented environment already populated with a fair amount of applications capability encapsulated as objects. It will be C++-based, and 32-bit-specific, but will run with either OS/2 or UNIX. Other versions should be easy to develop for any platform that supports a C++ compiler. (Asked how it differs from NextStep, outside its support for standards, Metaphor chairman-founder Dave Liddle candidly answers, "Not much. But we're four to five years further along from where they started." What can developers do now? Develop for OS/2 or UNIX, and use C++, he answers.)

"Marla Maples?" says Donald Trump. "She's just a good friend."
"GO? Metaphor?" says IBM. "They're just our good friends."

The combination of IBM on one side and Metaphor and Liddle on the other is promising. Liddle has garnered wide acclaim for his technical savvy, his understanding of user interfaces, and his articulateness in expressing them. He has also managed to remain on good terms with all the industry's increasingly polarized factions, and is widely perceived as objective. IBM is wise to team up with him, and to do so in a way that leaves him some independence. But the alliance will clearly compromise that independence somewhat. As a niche vendor, Liddle could afford to sit on the sidelines and be clear-sighted, diplomatic and objective. Now, he's moving right into the fray -- into the battle that will follow when people realize that the conflict among UNIX, OS/2 and DOS/Windows is not something to be won, but something to move forward from.

Geopolitics

Jim Cannavino, IBM vp and general manager of Personal Systems, and Joe Guglielmi, also IBM vp and president of the Application Solutions Division, will also sit on the Patriot partnership committee. The company is basically a financial entity that will own the intellectual property; development will be carried out by Metaphor (the majority) or IBM employees. IBM will sell all IBM-based implementations; Metaphor will market licenses or implementations for all other hardware. The relationship began in March 1988, when Guglielmi spearheaded IBM's investment in Metaphor, which resulted in a 386-OS/2 version of the Metaphor system, which until then ran on a 68000.

Interestingly, there's one more geopolitical note in this: Liddle is on the board of GO Corp., which also has a new operating environment. Theoretically, GO's is for the roving notepad, while Patriot's is for the sedentary desktop, but this is a difference that may not last. With luck, Liddle may foster some common object protocols, or he may decide that discretion requires him to remove himself from GO.
In the database world, SQL and portability and access to mainframe data (so-called "legacy" data) are all the rage, but they've been oversold. Aside from decision-support, what users often really need is access to mainframe applications, not just data. For example, from a new applicant-tracking system you might want to add a new employee to the payroll (see page 5). This is -- or ought to be -- more than a simple SQL update statement: There are probably a variety of security checks, ancillary transactions and other procedures involved. You don't want just access to the data; you want to use the applications, constraints and business processes that give it life and meaning. In other company, you might say you want objects...

Oddly enough, you can now get this capability from a relational database vendor providing access not just to DB2 but to all CICS database applications. Sybase's Open Server for CICS this by translating remote procedure calls from Sybase Net-Gateway into the appropriate syntax for CICS applications. Those RPCs are passed to it by Sybase's Net-Gateway, which handles calls from any front-end following the Sybase Open Client protocols, which now include not just Sybase front-end tools but a number of other applications such as 1-2-3 with DataLens. (Netwise recently announced Mainframe RPC, which is much the same idea but less database-specific; it provides the same foundation for interoperability but on a broader basis -- and with less of the database- and transaction-specific tools and translation facilities than Sybase's offering.)

Reverse engineering means you can rewrite old applications for a new platform (maybe with a few enhancements along the way). Portability means you can move old applications from one platform to another with minimum effort. True interoperability, or incremental engineering, means you can get new value from old applications right where they are -- by integrating them with new applications on new platforms.

The Open Server software includes both the runtime portion that translates the request into SAA-speak (LU 6.2 and the like), and tools for defining and specifying which mainframe procedures you want to use. That is, it isn't magic: A programmer needs to determine know which transactions are specified, and provide routines to convert the calls and parameters into the proper format for the individual mainframe application. What's the proper term for salary and pay period, in this context? How do you specify the start date, the job classification, the W2 status? (This is also a handy way for MIS to control access to "its" data.)

Open Server also handles static SQL, or compiled SQL statements, with Open Gateway for DB2. While almost everyone now has SQL access to DB2 and other relational databases, with ways to parse and pass on SQL queries, no one else yet provides easy access to static SQL -- or compiled SQL statements maintained by DB2. These are canned transactions that execute two to three times as fast as dynamic SQL.

Mainframes as servers, not hosts

Thus Open Clients fire off what amounts to a mainframe database transaction by sending a client request through Sybase's Net-Gateway and Open Server.
In essence, Open Server is encapsulating the mainframe transactions as procedures that can be called by an Open Client with Sybase’s remote procedure calls. This is akin to what you get with an object-oriented database: The methods come with the data. The difference is that the encapsulation is not supported by the environment, but rather is added on by Sybase’s tools, which know how to operate in CICS and to make a client request for a transaction look to the mainframe like something that came in over a 3270 terminal. It’s neat, and it’s an example of the interoperability — rather than interchangeability — of different kinds of databases mentioned above. Open Server could also be an interface to an object-oriented database as easily as, or more easily than, to a traditional database.

The value of Open Server for MVS is not intrinsic — that is, this might not be something you’d do if you were starting fresh — but rather practical: There’s a lot of business knowledge and invested work embodied in those old mainframe applications, and it would cost a lot of time and work and disruption to re-engineer and rewrite them. Why not just keep using them, neatly encapsulated?

A legacy animal gets the touch of life
We have begun early planning for the Fourteenth Annual PC (Platforms for Computing) Forum, which we will hold in Tucson from March 10 to 13 next spring -- two months later (and warmer!) than last year. The theme will be "Beyond the desktop: Networks and notepads." As a subscriber, you will receive registration materials shortly after Comdex, in December.

Topics will include: tools for groupware, E-mail-based, network and database applications; new object-oriented databases (see this issue) and operating environments such as GO's and Patriot Partners'; visualization of qualitative data such as text, priorities, workflows, data structures and program sequences; natural language and other text-based applications; virtual reality; as well as more down-to-earth concerns such as operating system wars, distribution channel trends, customer attitudes, and the changing marketplace. Many of these technologies will also be illustrated by demos from start-up companies and a few larger outfits dabbling at the leading edge.

---

COMING SOON

- Virtual reality.
- New groupware -- compare and contrast.
- Network navigation.
- Machine-assisted translation.
- The Douglas brothers -- Hofstadter and Lenat.
- MathSoft and Mathematica.
- Application servers.
- And much more... (If you know of any good examples of the categories listed above, please let us know.)

- ReLEAST: UNIX in the Soviet Union.

---

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave., New York, NY 10152; (212) 758-3434. It covers pcs, software, CASE, groupware, text management, connectivity, artificial intelligence, intellectual property law, object-oriented tools and other topics. Editor & publisher: Esther Dyson; associate publisher: Daphne Kis; circulation & fulfillment manager: Lori Mariani; executive secretary: Denise DuBois; editorial consultant and copy chief: William M. Kutik. Copyright 1990, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; but we gladly arrange for reprints or bulk purchases. Subscriptions cost $495 per year, $575 overseas.
The manifestos

"The Object-Oriented Database System Manifesto" is the work of Malcolm Atkinson, University of Glasgow; David DeWitt, University of Wisconsin; David Maier, Oregon Graduate Center; Francois Bancilhon, Altair; Klaus Dittrich, University of Zurich; and Stanley Zdonik, Brown University.

"The Committee for Advanced DBMS Function," which produced the "Third-Generation Data Base System Manifesto," consists of Michael Stonebraker, UC Berkeley; Lawrence Rowe, UC Berkeley; Bruce Lindsay, IBM Research; Jim Gray, Tandem; Michael Carey, University of Wisconsin; Michael Brodie, GTE Labs; Philip Bernstein, DEC; and David Beech, Oracle.

Copies of the manifestos are available from the people listed.

Fun with Juan and Alice

Alice: So you think OS/2 is catching up to UNIX?

Juan: Well, it seems they're announcing new versions of it every day.
# Release 1.0 Calendar

<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 2</td>
<td>&quot;The KGB, the computer, and me&quot; - your local PBS station. Billed as a &quot;comedy thriller,&quot; a NOVA documentary/reenactment of Cliff Stoll's hunt for a network intruder, as recounted in his book &quot;The Cuckoo's Egg.&quot; Stoll plays himself.</td>
</tr>
<tr>
<td>October 3-5</td>
<td>Seybold Conference - San Jose. Electronic publishing in all its guises. Call Kevin Howard, (213) 457-5850.</td>
</tr>
<tr>
<td>October 7-10</td>
<td>CSCW '90 - Los Angeles. Computer-supported cooperative work, with a slight (but lessening) academic flavor. Sponsored by ACM. Including a panel moderated by Lotus's Irene Greif, with MIT &amp; Agility's Tom Malone, Barry James Polsom, PW's Sheldon &quot;Notes&quot; Laube and Esther Dyson. Call Frank Halasz at (415) 494-4750, or Tora Bikson, (213) 393-0411.</td>
</tr>
<tr>
<td>October 8-10</td>
<td>Case users group meeting - Scottsdale, AZ. &quot;Making CASE work.&quot; Sponsored by Case Research Corporation. With Vaughan Merlyn. Call Debbie Boren, (206) 453-9900.</td>
</tr>
<tr>
<td>October 9</td>
<td>&quot;We know where you live&quot; [...] and a lot more] - your local PBS station. A Nova documentary about direct marketing -- as enabled by computers.</td>
</tr>
</tbody>
</table>

Release 1.0 19 September 1990
<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 14-17</td>
<td>EDUCOM '90 - Atlanta, GA. Sponsored by Georgia Institute of Technology. Speakers include: President Jimmy Carter, Steven Jobs, Robert Allen, Chairman &amp; CEO of AT&amp;T. Call Angela Walsh, (609) 520-3350.</td>
<td></td>
</tr>
<tr>
<td>October 14-17</td>
<td>Computer services and communications &amp; information systems seminar - Baltimore. Sponsored by Alex. Brown &amp; Sons. Call Sarah Hess or Molly Hogan, (301) 727-1700.</td>
<td></td>
</tr>
<tr>
<td>October 16-20</td>
<td>COMPFAIR 90 - Budapest, Hungary. Sponsored by IDG World Expo and Hungexpo. Contact: Zsuzsanna Lestak-Horvath at 36 (1) 1573-555 or fax 36 (1) 1285-034.</td>
<td></td>
</tr>
<tr>
<td>October 18</td>
<td>Massachusetts Computer Software Council's fall membership meeting - Boston. Call Joyce Plotkin at (617) 437-0600.</td>
<td></td>
</tr>
<tr>
<td>October 21-24</td>
<td>**EDventure East-West High-Tech Forum - Budapest, Hungary. Explore the problems and opportunities of high-tech business in Eastern Europe and meet your peers in a limited-attendance conference focused on contacts, not speeches. Sponsored by Edventure Holdings, with speakers and attendees from both sides. Call Daphne Kis, (212) 758-3434. By invitation only.</td>
<td></td>
</tr>
<tr>
<td>October 22-24</td>
<td>Electronic Messaging '90 - San Francisco. &quot;Beyond interpersonal communication.&quot; Sponsor: Electronic Mail Associa-</td>
<td></td>
</tr>
</tbody>
</table>
tion. With Helene Runtagh, GEISCO; Mike Zisman, Soft·Switch; others. Call Anne Spence, (703) 522-7111 or send e-mail via AT&T Mail: 1ema; CompuServe: 70007,2377; Dialcom: 63:PRD003; SprintMail: EMA; EasyLink: 62886257; iNet: ema.association; GEnie: EMA; Envoy 100: EMA; MCI Mail: EMA/2544290. (Do all these numbers make you long for fax?)

October 23
Sixth annual Eiffel user conference - Ottawa, Ontario. Sponsored by Interactive Software Engineering. Call Darcy Harrison, (803) 685-1006.

October 24

October 29-31

Oct 29-Nov 1

Oct 29-Nov 2
First annual meeting of the Soviet UNIX users' group - Moscow. Sponsored by SUUG; coordinated by the International center for scientific and technical information. Several hundred attendees are expected, including Bill Joy; representatives of the European UNIX Users' Group and usenix have been invited. Call Dmitri Volodin, 7 (095) 231-21-29, or V. Leonas, 7 (095) 120-69-21, or Esther Dyson, 1 (212) 758-3434.

Oct 29-Nov 2
AutoCAD Expo - Moscow. Sponsored by Autodesk and Parallel. Call Semyon Becker, 7 (095) 287-49-15 or 946-28-37, or Alison Fenwick, 44 (483) 303322.

Oct 31-Nov 2

Oct 31-Nov 2
MacroMind developers' conference - San Francisco. Call Scott Walchek or Amy Shelton, (415) 442-0200.

Oct 31-Nov 2

November 4-7
*ADAPSO management conference - Phoenix. Speakers include Steve Forbes Jr. and William Davidow of Mohr, Davidow Ventures. Contact: Ellen Kokolakis, (703) 522-5055

November 5-9

November 6-7

November 6-9
Software Development Conference - Boston. Sponsor: Miller Freeman Publications. Topics include logic processing with decision tables by Jim McCarthy; heresies of software design with P.J. Plauger. Call KoAnn Tingley, (415) 995-2472.

November 11

November 12-16

Release 1.0 19 September 1990
<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 12-16</td>
<td>Neuro-Nimes '90 - Nimes, France. Neural networks and their applications. Sponsor: Association for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cognitive Research. Contact: Annick Crozet, (331) 47806452 or (331) 47807000.</td>
<td></td>
</tr>
<tr>
<td>November 13-14</td>
<td>Outsourcing systems operations: revolution in information systems management - Vienna, VA. Sponsored</td>
<td></td>
</tr>
<tr>
<td></td>
<td>by INPUT. Call Richard Ferrotti, (703) 847-6872.</td>
<td></td>
</tr>
<tr>
<td>November 14-16</td>
<td>Pacific Rim international conference on artificial intelligence '90 - Nagoya, Japan. Sponsored by</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the Japanese Society for Artificial Intelligence. Special sessions for AI in engineering and AI and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>large-scale information. Call Shigero Sato at (813) 479-5535 or fax (813) 479-7433.</td>
<td></td>
</tr>
<tr>
<td>November 15-16</td>
<td>International protection of intellectual property: current issues and directions - San Francisco.</td>
<td></td>
</tr>
<tr>
<td>November 27-28</td>
<td>The 1990 Forrester Fortune 1,000 technology forum - Cambridge, MA. Client/server computing: the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>impact. Sponsored by Forrester Research. An all-star line-up of speakers including Scott McNealy,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>James Cannavino, Andrew Grove, Jim Manzi, Michael Hallman, Charles Exley, Jr., James Treybig.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Call Jean Baranowski, (617) 497-7090.</td>
<td></td>
</tr>
<tr>
<td>November 26-29</td>
<td>Conference on software maintenance - San Diego. Sponsored by IEEE. Call Michelle Carbone, (202)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>371-1013.</td>
<td></td>
</tr>
<tr>
<td>December 3-4</td>
<td>*Emerging technologies conference - Innisbrook, FL. For developers eager to use the latest in</td>
<td></td>
</tr>
<tr>
<td></td>
<td>technology -- object-oriented programming, cooperative processing, IBM repository, notepad</td>
<td></td>
</tr>
<tr>
<td></td>
<td>computing, image management, etc. Sponsored by ADAPSO's Technology Information Services Committee,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Industry Division. Call Phyllis Cockerham, (703) 522-5055.</td>
<td></td>
</tr>
<tr>
<td>December 3-5</td>
<td>European Publishing conference - The Hague, Holland. Sponsored by Seybold Limited. Contact: Laurel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Brunner, 44 (323) 410561 or fax 44 (323) 410279.</td>
<td></td>
</tr>
<tr>
<td>December 3-7</td>
<td>Toulouse '90 - Toulouse, France. Software engineering and its applications. Sponsored by EC2.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Contact: Jean-Claude Rault, (331) 47807000 or fax (331) 47806629.</td>
<td></td>
</tr>
<tr>
<td>December 5-8</td>
<td>CASE '90 - Irvine, CA. The fourth international workshop on computer-aided software engineering.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sponsored by Index Technology, IEEE and several academic institutions. Call Ron Norman, (619) 594-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3734.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Call Marilyn Reed, (800) 635-5537 or (203) 226-6967.</td>
<td></td>
</tr>
<tr>
<td>January 28-31</td>
<td>Network Computing Forum - Washington, DC. Sponsored by World Expo Corp. Call Christine Krajewski,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(508) 820-8126.</td>
<td></td>
</tr>
<tr>
<td>February 12-15</td>
<td>*Software Development '91 - Santa Clara. Sponsored by Miller Freeman. Call Lynne Mariani or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Angela Hoyte, (415) 995-2471.</td>
<td></td>
</tr>
<tr>
<td>Feb 23-March 1</td>
<td>Graphicon '91 - Moscow, USSR. Sponsored by USSR Academy of Sciences. Call Kathleen Milles, (312)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>644-6610.</td>
<td></td>
</tr>
<tr>
<td>March 3-7</td>
<td>*Seybold Seminars '91 - Boston. Electronic publishing's premier conference. Call Kevin Howard,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(213) 457-5850.</td>
<td></td>
</tr>
</tbody>
</table>

Release 1.0 19 September 1990
March 10-13  **EDventure Holdings PC (Platforms for Computing) Forum - Tucson, AZ. "Beyond the desktop: Networks and notepads." Sponsored by us! Registration materials will be mailed to subscribers after Comdex; two paid attendees per subscription. Contact: Daphne Kis, (212) 758-3434. See page 21.


March 26-28  Spring symposium series - Stanford University. Sponsored by The American Association for Artificial Intelligence. Call Carol Hamilton, (415) 328-3123.


May 20-23  Spring Comdex - Atlanta, GA. Sponsored by The Interface Group. Call Elizabeth Moody, (617) 449-6600.


June 9-16  Poznan international fair - Poznan, Poland. US exhibits sponsored by Department of Commerce, Eastern Europe Business Information Center, 1 (202) 377-2645.


Please let us know about any other events we should include. -- Denise DuBois

*The asterisks indicate events we plan to attend. Lack of an asterisk is no indication of lack of merit.
Please enter my subscription to Release 1.0 at the rate of $495 per year in the U.S. and Canada. Overseas subscriptions are $575, airmail postage included. Payment in U.S. dollars must be enclosed. Multiple-copy rates on request. Satisfaction guaranteed or your money back.

Name
Title
Company
Address
City State Zip
Telephone
How did you hear about Release 1.0?

Please fill in the information above and send with your check payable to:

EDVENTURE HOLDINGS INC.
375 PARK AVENUE, SUITE 2503
NEW YORK, NY 10152

If you have any questions, please call us at (212) 758-3434.

Daphne Kis
Associate Publisher