OBJECT MARKETS

The notion of electronic markets for class libraries and objects is appealing. After we gave talks about it in London and San Francisco this month, two separate start-ups came discreetly forward with business plans, while a number of other people expressed less formulated interest. Meanwhile, Autodesk subsidiary American Information Exchange, the talk of the show at Object World in San Francisco, has shifted its priority from selling information to selling objects. Trilogy Development Group is developing technology that could support a highly advanced form of such a market. And guru Brad Cox has outlined how objects could monitor their own use and remit payment information automatically from users to a clearing house.

The promise is that an efficient, global-network-based market for objects would maximize utility and would offer market access on equal terms to small vendors who are now hampered by the competitive advantages of giants. However, there are a lot of obstacles to achieving such a seamless online worldwide electronic marketplace. Not least of these is that distribution of objects has never been the difficulty; the value of a market has more to do with the visibility of information than with the physical (or electronic) exchange of goods.

In the end, the challenge and the promise lie more in electronic management of the proliferation of information about objects than in electronic delivery. Marketing, brand names and persuading the customer to buy will still matter more than baud rates or response times. Although electronic communications will flatten the market somewhat and reduce the power of current leaders such as Microsoft, we won't have a "democratic" market. People will still buy from established vendors, and new information brokers with reliable knowledge about objects and their appropriate use will gain market power. Equal opportunity does not lead to equal results.

The premise of this issue of Release 1.0 is not that object-oriented technology leads inexorably to electronic markets for software components, but rather that technology and business infrastructures are co-evolving to a world of electronic markets. Many of the software components for sale won't fit classical definitions of object-orientation, but they will serve to foster the transition from software development as fabrication, to a broader community that accomplishes system/application development as assembly. Current information

PC FORUM 1993: 21-24 FEBRUARY
services are moving from reselling passive information that a reader inter-
prets or manipulates, to selling active, executable classes, objects or
components that a designer/developer assembles and a user uses. (Ideally,
they can synthesize/cooperate by themselves.) Basically, we are looking at
the convergence of technological and commercial objects. Someday we’ll
wake up and see how well-suited they are for one another, and we’ll have a
market accompli. In this newsletter we discuss the likely dynamics of such
electronic markets. We also cite a number of early steps in this direc-
tion, from both the OOPS and the electronic markets communities.

The market: Objects as artifacts

Object-oriented technology requires the identities and functions of code
modules to be explicit -- and thus potentially marketable. They are the
most technologically-determined leading edge of a far broader field of
software/information components that can be distributed electronically, or
electronic goods. Software components that aren’t strictly speaking ob-
jects or classes, but that will still let users assemble systems, include
items such as fonts, DLLs, subroutines and even applications. Electronic
goods also include data, text, images, diagrams and other information.

Thus objects aren’t simply technological artifacts; they are potentially
commercial items with identities, functions, interfaces, and perhaps even
pricing. These aspects of objects are perhaps even more important than the
code that implements them.

-----------------
There will be objects for markets, there will be markets for
objects, and there will be electronic markets. Ergo, there
will be electronic object markets.
-----------------

A cycle of positive feedback should start: A broader market for objects
makes it feasible for more people to build them for resale and to sell them;
market forces can also be harnessed within corporations, so that more and
more internally developed classes can be marketed outside. Our culture is
beginning to recognize that intellectual property is a commodity like any
other. The difference is that object-oriented classes and objects have
unique capabilities and requirements that make object markets especially
valuable -- and especially challenging to operate effectively. Market
forces will foster the development of well-defined, interoperable objects
that adhere to broad standards rather than objects designed to be re-used
only within a single organization.

Sellers: A new world order

Object orientation allows us to build software out of fungible components
instead of creating software monoliths that are difficult to modify or ex-
tend. This modularity changes the business dynamics for suppliers of these
components. Application vendors are getting involved in setting standards
for some of these modules, most recently mail, taking over some of the ter-
ritory currently held by OS vendors. Cross-OS portability will become more
and more common.

But interoperability, everyone’s goal, has a flip side: interdependence.
Foundation classes -- those that provide basic objects such as data struc-

Release 1.0 32 July 1992
turers (strings and dates), memory management or math -- will likely take on some of the current role of operating systems as the defining platforms for most application classes. For example, if you use Borland's foundation classes for your basic system objects, your higher-level objects will inherit from those classes or will at least contain them, and you will look for corresponding classes for functions such as asynch communications, linear programming and so on.

Contenders in the foundation-class field (will) include Borland, Symantec/Apple with Bedrock, IBM/Apple with Taligent, NeXT, Microsoft, ParcPlace Systems and Digitalk, Rogue Wave, Sun and a host of object converts. (You can develop foundation classes without owning an operating system company, but it's clear every OS vendor wants to be in this new game.)

However, the total "contribution" of a foundation class to a fielded application is smaller than that of an operating system; many erstwhile operating system components will be handled by higher-level classes. Besides, you can usually automate the necessary conversions from one environment to another, and there are various ways around the problems of compatibility. Higher-level classes, such as database and communications classes, to say nothing of application frameworks, will be designed to work across foundation classes (and OSes).

We believe Microsoft's current ascendancy is an anomaly, not just for Microsoft but for the computer business as a whole. Object-orientation means that it's easier to mix software from various sources. All the interfaces are cleanly defined, so that third parties can add in their own wares. Yes, there will no doubt be a few leaders, but it's unlikely that a single vendor will hold sway the way Microsoft does today. The computer business, once a discrete industry serving other industries, is now becoming more and more integrated into the business world as a whole. Everyone uses computers, everyone buys them, and more and more firms sell software as an adjunct to or vehicle for whatever else they sell. The value-added in computers is software, and the value-added in software is application- and market-specific knowledge.

Thus, although there will always be natural coalescences of market influence around strong vendors, it will be harder for any single vendor to gain (or retain) a stranglehold over the industry. OS vendors will lose their power -- but not so much to individual application vendors or even foundation-class vendors as to users and resellers. Object-orientation will allow customers to acquire mostly coherent, cohesive systems from a fragmented supplier community. This time around, the industry is working hard to limit its dependence on anyone!

If all this reduces the power of, say, Microsoft, who will take over? As we've shown, foundation-class vendors will hold some technological sway but no one vendor will have the power to control most of any substantial market.

As hardware, operating systems and even class libraries become commodity objects, then, the only asset with more-than-commodity value may be usable in-
formation services about software offerings: That is, which objects to choose, not the code in the objects itself.

Thus, market power will move to the intermediaries -- the people and firms who know what's available and can link buyers and sellers; object brokers and system integrators who can validate, assemble and customize systems out of components. Currently that looks like consulting firms: Arthur Andersen, say, or Perot Systems or Tangent International (page 20). (Most large computer and software firms have their own consulting arms, but they suffer from at least the perception of bias.) These firms are in the business of knowing what software is available and how to use it for their customers.

But history still counts!

Of course, object brand names will still matter tremendously, as they do in any market where differentiation is important. It's hard to specify exactly what quality is -- certainly it's more than the absence of bugs -- so you have to rely on brand names or familiarity. Some hypothetical examples:

- Apple always seems to come up with clever ways of doing things.

- Microsoft doesn't get it right the first time, but it sticks around to fix it. (Why didn't DR DOS sell better?)

- That little company, Juan & Alice Classy Classes, just seems to understand exactly how we like to do things in our company. They subclass the (hypothetical) Federal Accounting Standards Board Class Library just right. The FASB class library is provided as a nonprofit service to member accounting firms and to software vendors who are free to (re)use it. It comes in a variety of flavors to work with system-class libraries from the major system software vendors: Apple, Microsoft, Novell, Symantec, Sun, UNIX Software Labs and oh yes, IBM.

- Lotus has some foundation classes for managing text databases and mail that benefit from its success with Notes, while a variety of vendors sell almost indistinguishable databases that work both to manage data and as installable file systems/object brokers interoperating with other system objects.

Buyers: The promise of reuse

The trend to object-oriented technology is now being driven not just by the availability of C++ and Smalltalk, but by changing customer needs and the sheer cost of developing robust tested code. Economics is driving us toward reuse not to reduce the costs of development but to save the cost of re-development: We want to keep using what has already been done with whatever is new. In the past, software was generally built in-house as an undifferentiated mass/mess for one-time use; though the software was executed over and over, the designs and processes it embodied could be abstracted and modified only with difficulty and generally weren't used elsewhere. That ensured the programmer a full-time job ever after.

As more people use software, companies want faster ways to develop and modify systems, using existing applications and data and the knowledge embedded in them. Unfortunately, this original installed base wasn't created with
reuse in mind. This commercial impetus is reflected in many moves towards the basic benefit of OOPs: reuse. Reuse, in the broad sense, is nothing new. It can be achieved not just through pure object-orientation with the full underpinnings of inheritance and dynamic binding, but also through modularity, client-server architecture, documentation, templates, customizable applications, scripting, standards -- and quality that makes code worth reusing in the first place. In the extreme, reuse starts with copying software and having more than one user use the same code -- sort of the reductionist version of reuse. All these are on a spectrum.

In the end, reusability isn't yes or no, it's a question of degree, scale and granularity. At one end, you're assembling large-scale modules: Does this client application work with that database? At the other end, you're writing code, or reusing the words of a language. With object-oriented inheritance, you can reuse implementations not just whole, but you can inherit behaviors or structures to reuse in building new objects. The benefit is not just in the cost of making the object, but in the cost of using it. Not only do you not have to rebuild it, you do not have to redefine a familiar object, redocument it or train people how to use something new.

------------------
What makes objects special from a commercial point of view isn't inheritance or encapsulation or polymorphism. It's that they have discrete identities and explicit interfaces -- or relationships to other objects. They define themselves and they specify the interfaces and protocols they use. (From the customer's point of view, the technical aspects make them usable and useful once they are purchased.)

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

Reuse in many flavors

Fundamentally, there are three levels of reuse: by programmers, by assemblers, and by users.

Reuse by programmers (fabricators) is where the technical aspects of object-orientation come into play: inheritance, etc. The reuser actually gets in there and writes code, adding features or modifying existing objects and creating new ones. This user needs well-defined objects so that he can know what he's working with, and well-built ones so that changes are easy to make. The programmer typically works with source code, and modifies all but the most basic foundation classes.

-------

1 Client-server architecture allows many client applications to "reuse" not just the data but the applications, constraints and logic on a server; client-sever means more than just "database-based." A server can allow the shared use of communications facilities, text bases, specialized computing tools (hardware or software), etc. As vendors such as Sybase, Lotus and the OODB crowd make clear, by putting logic on the server you can be sure that it's used consistently and applied to the appropriate data each time it's used. The parallels to putting methods/procedures with the data in an object are clear.

Release 1.0 32 July 1992
This model of reuse is anathema to some people, since it involves all the "mistakes" of current programming. It basically involves rewriting code -- and introducing modifications and potential mistakes. Says Brad Cox, who equates C++ with fabrication technology and Smalltalk to an assembly technology: "Silicon fab lines (i.e. fabricating objects) and soldering guns (i.e., assembling encapsulated objects) are not equivalent hardware technologies!"

**Reuse by assemblers** is where the leverage and the commercial issues come in -- and where we're focusing. Instead of writing code, this person assembles modules -- which may not even be "truly object-oriented" inside. Encapsulation, not inheritance, is the germane aspect of object-oriented technology. The user's task is not to be a creator in the fine-grained sense, but rather to combine existing components to create something new. He does not muck around with an object's source code, but he may write additional code to add functionality to or around existing objects -- just as a pc-maker assembles ICs, to use a popular analogy.

Objects should match their functions and real-world models just as any software should. Objects get close to the real world just as applications do, and though they are modifiable and recombinable, they should be structured properly from the start. For example: A person is easy to subclass into different kinds of people, but it's hard to subclass an employee into a customer. Is an address an object or an attribute? An affiliation? It's simple if all you want to do is mailing lists. But then: How many subscribers work at this institution? What if the company moves or starts a new branch with a different address?

In order to assemble objects effectively, a user needs to be able to find, assess and combine appropriate objects. Helping users to do so is the basic task of the object market.

**Reuse by end-users** is where "cute" objects such as screen icons, menu-builders and other simple objects come into play. Reuse by users also occurs at the level of application integration, with scripting, drag-and-dropping and the like. Is a user doing mail-merge reusing a database and a word-processor? Arguably. What if he writes a script to combine a database object and a word-processor object to generate letters automatically each Monday? The use of the word doesn't matter; the trend is clear.

**Making choices**

Note the distinction between fabrication, where objects can be subclassed and implementations inherited, where the code within them is reused, and assembly, where the object as a whole is reused in the black box or software IC model. Terminology varies, but generally we'll assume that programmers reuse classes, while assemblers reuse objects. In general, C++ is favored by the fabricators, whereas objects designed for assembly are favored in the Smalltalk world. Still, Smalltalk visionary Adele Goldberg of ParcPlace is adamant that assemblers should be able to look at source code and learn from it, even though she generally favors the assembly model of reuse. (That's the difference between encapsulation and information-hiding.)

As with "regular" software, developers can take a variety of approaches to development: How much work (value) does he want to add to off-the-shelf
components, and how much does he want to reuse? These are all the same kinds of questions a customer faces today: customized application or database tools? A text database or a relational database with mail-merge? Will accounting and customer support use the same customer database? (We hope so!) Will marketing have access to it too, to see which products are selling? Where are customer complaint files maintained? Can they be easily inspected both by product managers and by customer support people?

The major restriction on any customer's choices is generally what's already in-house. Yes, object-orientation gives you the freedom to modify and extend, but you still have to modify and extend what you've already got. Just as you have to build software on top of a specific hardware platform, you have to build object-oriented systems descended from a particular foundation class library, whether you do so directly or as an assembler. This refers both to technical issues such as operating systems and networking protocols, and to ostensibly simpler but still dangerous aspects such as the terms you use for "employee" or the particulars of a parts-numbering scheme.

Supply side: The object author/publisher

The object publisher is key in this exercise: He's the source of supply. In theory, object-orientation will unleash a flood of market-ready objects. If proper procedures are followed, internal shops will craft a set of carefully engineered, well-documented objects. All you need is a price list and an advertising blurb...

But reality is far different. Aside from companies devoted to selling software, and consultants devoted to selling information, few companies are reselling objects or classes (with the notable exception of Texas Instruments, which is exploring the idea). After all, how many other goods built for in-house use could be sold on the outside? Take the case of Infinity International Financial Technology, a 25-person software house in Mountain View that builds soup-to-nuts object-oriented financial applications and tools for international banking, using C++, Sybase and a set of foundation classes. It built its own database class library because it couldn't find what it wanted from a third party. But what it built was for internal use, not a product with documentation for general sale. Says founder Robin Vasan, "For the time being we need to concentrate on marketing and supporting our complete financial class library and applications to top-tier financial institutions; packaging generic classes would be a diversion of our resources." Recently the company has seen considerable interest in its database classes; rather than sell and support the technology directly, it's considering the possibility of licensing it to a third party for resale.

So yes, it will happen, but both corporations and individual developers will discover that selling objects is more than making up a price list and contacting a reseller -- even once the markets limned here are in place.

The object republisher/value-added reseller

As noted above, the intermediaries who supply classes and objects to their reusers, along with ancillary services, will play a key role in the object markets of the future. Their tasks include (to varying degrees):

- finding appropriate classes/objects for their markets.
- negotiating terms with their suppliers, including reuse rights, warranties, support and update commitments and other specific terms and conditions.

- testing the software and validating the authors' claims of performance, compatibility and functionality.

- providing support; even though the original developer may provide some support, the reseller should be able to provide support to a customer using classes from several sources.

- collecting payments from customers and remitting them correctly (minus a cut) to copyright-owners.

Product definition

Basic information about objects for sale sold must come from their creators (although generally a reseller must validate it). This means everything from a functional definition, to specific data -- protocols and interfaces; inheritance hierarchies; complementary tools, objects and class libraries; platforms or databases supported, etc. It could include useful ancillary information such as diagrams of the procedures performed or examples of capabilities. All this is the same as software documentation in general; object-orientation makes it more important and requires it to be explicit. This explicitness is a benefit, since it affords the possibility of automated searches for complementary objects and classes, and compatibility management (cf. Trilogy, page 16). You could also search this information with text tools -- for the functional description -- and then apply more complex structured queries to match desired attributes such as works-with, requires, uses or replaces certain other components. Softer information is important too: Is the author or vendor well-known? Who else has used these products? How effectively do they match business processes?

Selling software and especially objects has a nice built-in impetus for customer loyalty: The customer's needs depend on what he already has, far more deeply than for most products/services. Because of inheritance and the structure of classes, a customer can purchase a new child object and already have the superclass to inherit from. This is both a technical and commercial fact. It's akin to knowing what database your customer has so that you can sell him appropriate applications, but on a much more fundamental level.

Technically, objects may all be able to send messages to each other, but they have to send the right messages. Objects make it easier to define the interfaces, but the issue of complementary interfaces does not go away.

Third-party knowledge

Just as important and more difficult to assemble and manage is cross-vendor knowledge -- solid information on which vendor's software works effectively with which others', and how. There is a huge combinatorial explosion of information that becomes a huge burden to test. For example, if you're using a certain accounting system, certain payroll systems are best suited to work with it. A second issue is that you generally can't just load a couple of objects and go (claims to the contrary notwithstanding); you have to let the rest of the system know that they're there. That involves a
certain amount of installation, configuration and other technical effort and knowledge. That's why we see third-party intermediaries as playing such an important role in the object market of the future.

Validation and quality control

Validation and quality control is another tough practical problem. Like the issue of interoperability, it will shift the market towards concentration on a limited number of brand names or known republishers, at least for the foundation class libraries. Within vertical markets, specialty suppliers (or industry players reselling technology) may gain market stature. If your parent classes are thoroughly tested, you may be more willing to go with the modifications/subclasses of an (initially) unknown industry specialist.

Payment policies

The final, fundamental issue is how these components will be charged for, especially as they get reused, incorporated into other components and otherwise lose their identity on their way into users' systems. What distribution channels and payment mechanisms will we use? What does this mean for the definition and protection of intellectual property? Some big vendors such as DEC are looking at getting into the information utility business, where they will build systems themselves and charge users for transactions. How will they compensate the originators of the components they use?

The ideal payment policy is like the ideal class library: It doesn't exist. Vendors and users will all have their own preferences. Even for current software, there is a variety of policies: per-user, per-server, site license, per-transaction, by time, runtime copies, OEM or bundling deals, maintenance, etc. In each case, there are trade-offs in record-keeping overhead, value received vs. cost to deliver, and so forth. (Should a wedding dress be cheap because it is used only once? Or expensive because it is made to order and has a high value to the customer?)

There is no single way to price "fairly," so pricing policies as well as prices will be determined in a trial-and-error, evolutionary way as a variety of schemes are tried out. Adding in tiers of distribution and reuse only complicates things: Resellers and reusers have to negotiate with the owner or licensed reseller of the copyright (or "useright"), and then keep careful records so that the appropriate royalties can be paid to the right person. Some vendors -- such as Rogue Wave now or Serius originally -- virtually give away their foundation libraries in hopes of building up an aftermarket. Companies like Cygnus don't charge for software at all, but make their revenues by providing support and consulting (see Release 1.0, 4-90). In general, the class library, C++-oriented vendors seem to approach their property like a language -- a one-time fee for unlimited reuse, while the object vendors are more comfortable with runtime or per-user fees. (But that's a gross generalization.)

Payment and distribution systems

Then there's the question of record-keeping and payment delivery, even once the policy is set. Consultant Brad Cox and Ryoichi Mori of Japan's Electronics Industrial Development Association (JEIDA) have posited the elegant idea of "superdistribution" of electronic goods that can monitor their own

Release 1.0 32 July 1992
usage: Each object will have method (or a wrapper with a method) to keep track of how it is used. The object then send reports to some payment-clearing facility or credit-card company which credits the account of the copyright-holder or his agent. The reports concern usage; the clearing facility determines how to convert that into actual charges. "Usage" means use by other applications as well as by human users.

The beauty of the concept is that it saves the whole bother of distribution middlemen and complicated ordering/delivery systems. The electronic goods can be freely distributed, just like shareware or freeware. Payment would depend on use, not copying, so that copying would become a positive act rather than one fraught with violations of rights. Philosophically speaking, this idea of superdistribution more accurately reflects the way information should be: freely available, yet it remits rewards to its creator when it is used.

"Whereas software's ease of replication is a liability today, superdistribution makes it an asset. Whereas software vendors must spend heavily to overcome software's invisibility, superdistribution thrusts software out into the world to serve as its own advertisement. Whereas the personal computer revolution isolates individuals inside a standalone personal computer, superdistribution establishes a cooperative/competitive community around an information-age market economy."

--- Brad Cox, Information Age

Of course, users will certainly be able to disable the reporting mechanism. On the other hand, software use should become cheaper per unit as costs are spread by the broader market likely with such a scheme, reducing the incentives for illegal use of each individual component. Cox and JEIDA suggest the use of some kind of hardware device -- computers equipped with a special chip that manages the reporting process, and requires it to work in order for the electronic goods to be used. JEIDA is working on such a chip.

Separately, Cryptologics, founded by National Semiconductor founder Peter Sprague, is about to launch a metered information-delivery system that can work with electronic information in any medium. Cryptologics, moreover, is working on particular markets rather than a global scheme. It monitors delivery and collects payments for established with information publishers who already sell information (currently mostly legal, scientific and technical information) to targeted customer bases. It monitors delivery of this information, which is encrypted, with a hardware device that includes a decrypter and a meter. Of course, once the user has downloaded the information he's able to do what he wants with it -- although not legally. (By contrast, the point of superdistribution is to equip all computers with such metering devices: the electronic goods can move around free because they always report on their usage.)

Distribution is not marketing

But the "superdistribution" concept does not eliminate the need for information middlemen. The objects will still need visibility, whether through advertising, Cryptologics' partners' targeted marketing, or paid rating or referral services such as Juan & Alice's Object Best-Buys.

Release 1.0	 32 July 1992
This problem has faced many ventures in electronic distribution of software since the early Eighties (see Release 1.0, 11-83): The "distribution bottleneck" is minor compared to the visibility bottleneck. That is, people aren't looking for software they can't get; they're looking (possibly) for things they don't know about. Merely making software more easily available won't solve this problem. For example, InfoNow, the latest of the electronic distribution start-ups, is offering a product-information service as much as a distribution service. Product information and some products are delivered to subscribers on a CD ROM; if you like a particular product, you can either download it after calling InfoNow for a software key, or you can order it through InfoNow for delivery the "normal" way, depending on InfoNow's contract with the supplier.

Cox suggests that software could advertise itself (as in InfoNow, but not really!), but to the extent that such an idea is successful, customers will be flooded with self-advertising software. (See Release 1.0, 6-91.) They will either use filtering tools or services (including consultants as well as software) to find what they want, or restrict themselves to looking at "sponsored" software, where the vendor has paid some medium to be highlighted or passed some quality hurdles. The software will highlight itself, yes, but not enough to overcome the noise of the net.

For the superdistribution scheme in particular, how can you persuade people to buy the requisite computer-meters or add-in chips? There would have to be something compelling about the meterware to persuade people to adopt it. This is the kind of scheme that doesn't happen because no one has a compelling interest in making it happen. \textit{It would be nice if} all the vendors decided to switch to meterware, but why should they? Large vendors are already distributing their products effectively to everyone who wishes to pay for them, while only the smaller players might take a risk and upset existing channels to reach a broader market and find new customers.

Finally, corporate customers will resist such decentralized distribution and purchasing because they would lose control of their budgets.

Nonetheless, we do look forward to watching the first successful vendor with a hot new application class use such a system (without the hardware assist). It will probably just have a routine that asks if you'd like to purchase it, takes your name and credit information and then automatically dials an 800 number to record a charge against your credit card. (Shareware is getting close to this now.)

More likely, we'll see an extension of current techniques, with users paying for classes or objects once. They will then pay for updates, modifications add-on subclasses and support.

\textbf{-------------}
\textit{If a reusable object sits in the forest and no one uses it, is it really reusable (or useful)?}
\textbf{-------------}

\textit{Release 1.0} \hspace{1cm} 32 July 1992
SOME EARLY INSTANCES -- PROTOTYPES, EVEN -- OF OBJECT MARKETS

There won't be a single marketplace for objects; rather, there will be many competing marketplaces and submarkets, just as there are for most other goods. Thus markets for objects will evolve just as the technology of objects does, with users of markets (buyers of market services) sending signals to market providers (sellers of market services). They will compete on the basis of audience, goods bought and sold, share of total market, quality and completeness of information, default payment policies and procedures and other criteria. Likewise, today we have markets operating through the pages of magazines, and others oriented around direct sales. Informal freeware and shareware markets already operate partially over electronic networks, and partially through networks of friends. There's some overlap, of course.

Markets may also specialize in, say, financial applications, NextStep classes or chemistry. Then there will be cross-market arbitrageurs, who may offer to help vendors enter new markets, or help buyers to find objects in markets they don't normally frequent. The current range of markets is the primordial soup from which tomorrow's more widespread, better-tuned markets will arise.

The same rules apply in electronic markets as in other ones.
As a distribution medium you get channel-rental fees; it's only as a selective remarketer, with the responsibility and contingent liabilities of support and product certification (explicit or implicit), that you get to earn a commission.

American Information Exchange: Object central

American Information Exchange was founded in 1986, with the goal of offering a marketplace for buying and selling information. (See also Release 1.0, 7-89, 11-91 and 12-91.) It has gone through several iterative redefinitions itself: Like a lot of Valley companies, it was as much a social experiment as a business. Autodesk invested in 1988 for a controlling interest, but pretty much left the company alone. (You could call that lack of interference or lack of support.) Last fall founder Phil Salin died, leaving a strong team and a powerful vision but something of a leadership hole.

AMIX's original concept focused on market research and consulting services -- implicitly embracing both canned and custom information. Its original marketing strategy of "go after customers likely to have pcs already" focused on information about the computer/software market. This may seem a little inward-focused, but it makes perfect business sense.

Now the company has moved one step further in this pragmatic direction, focusing more on sales of objects and information about them, based on the relative success of its various offerings. Its Smalltalk market has been the most successful in terms of the value of the items offered and the intensity of interest (although Computer Trends and Analysis has more documents). The obvious reason, in retrospect, is that you can easily get market research and consulting elsewhere (albeit usually in bigger chunks or by subscription only), while individual objects are harder to find. Also, electronic distribution is ideal for objects, whereas it makes less sense for
research reports (unless they're largely data). Its two object markets, for Smalltalk and C++ objects separately, are run by market-maker Agoric Enterprises, below. Although it's still in its infancy, AMIX is a model for a substantial sector of the market for objects.

Recently, the promotion budget has increased, courtesy of Autodesk. (New Autodesk cfo Eric Herr once worked at McGraw-Hill, and understands information markets from broad experience and a degree in economics.) That enabled AMIX to exhibit at the recent Object World in San Francisco, where it was the talk of the show and seemed to exemplify (although not precisely!) the point of Brad Cox's opening speech. The AMIX booth attracted 20 percent of the 5000 show-goers, estimates co-founder Gayle Pergamit, who is still sorting through the leads.

The company has been rolling out its services over the last few months, but it has had a more difficult time ironing out operational issues than it expected. A lot of users love the system in principle, but found early versions annoying to work with (ourselves included). In the background, the company and its various market-makers have been working to attract a critical mass of information/object suppliers so that the shelves will be full before the public is invited in. "It has been an exciting bootstrapping time for us," says Agoric co-founder Howie Baetjer. "Object World was really the first moment we felt ready to invite the public in" -- and Object World is still a pretty limited public!

AMIX makes no claims to provide support, quality control or other market management itself, although it tries to ensure that the market, market-makers or the vendors themselves take care of it. As the name says, AMIX is an exchange on which third parties manage markets. However, as a business matter, AMIX does pay attention to the quality of market-making and management, using common-sense criteria: Are buyers and sellers happy? Is the volume of business satisfactory? Are the appropriate items being bought and sold? (In this start-up phase, however, several of the market managers are AMIX employees, doing a job which AMIX may long run cede to third parties. Then the market managers may themselves compete: "My Classy Objects Boutique is better than his Objective Classes Corner!")

AMIX's answer to questions about quality control, validation and the like -- based on principle -- is "let the market do it." That should happen in a couple of ways, still using AMIX or other exchanges as a vehicle: Customers may prefer to buy from a publisher/reseller such as Metric or TCL (below), rather than buying direct (over AMIX) from unknown sources; these resellers should find AMIX a congenial place to do business. Other third parties may simply go into the business of supplying information about objects over AMIX, without actually selling the objects themselves. Customers can also check a vendor's references over AMIX; the one thing AMIX does prohibit is fraud or misrepresentation. Problems are brought to its attention by customers, and then it arbitrates disputes.

Agoric Enterprises Inc.: Market lab

Agoric Enterprises grew out of the Agorics Project at George Mason University's Center for the Study of Market Processes. A team led by Don Lavoie, professor of economics, got close to AMIX while doing a study on obstacles
to the creation of markets for software components. "We could envision all these gains from trade that aren't being realized because sellers can't find buyers," says Howard Baetjer, now a principal of the three-person company, and executive director of the Agorics Project in his day job.

AMIX offered to let the research team start a Smalltalk components and consulting market as part of their research, supplementing a C++ market run by a programmer with a Rolodex -- a great programmer who decided he was not a market manager. The experiment turned serious, and last April the group created Agoric Enterprises Inc. Its only current client/activity is the AMIX business, but it hopes to grow beyond that eventually.

Going into Object World, Agoric's AMIX Smalltalk market had about 40 items for sale and 20 consultants offering services; those figures should now increase significantly. Agoric is now talking with a variety of suppliers, including a large number of new leads from Object World. Although the motivation is to make money in part -- as the sign of a healthy business/market -- Baetjer and his team are also fascinated by how the process works. Right now, one of the trio's projects is to investigate extending the default license agreement AMIX supplies: "We'd like to offer a number of licensing options, concerning reuse, royalties and the like. The default AMIX agreement is too flat: 'The buyer is entitled to own-use only of anything purchased, and may not resell.' We'd like to take the lead on some of these issues, consciously building supporting institutions and market policies that people could get jazzed about."

Another issue is labeling and definition: Although it's confusing to users, it's interesting to note that the items for sale are not differentiated by the system as objects, classes, documents or simply electronic pointers to things that could be delivered physically. One of Agoric's current goals -- both business and research -- is to establish useful, clear conventions for labeling items, not just as to type, but what platforms they run on (or parent classes they inherit from), whether they have been reviewed and by whom, what documentation and support are included, and so forth.

That is an even more important issue -- but a tougher one to handle -- in the market for consulting services. So far, AMIX provides a structured form of e-mail for negotiating specs and conditions for custom-built objects -- or for consulting. Agoric hopes to foster the evolution of a richer, more useful set of framework specs and contracts, and draw customers by the efficient, visible market it will provide.

The period when just one or two vendors can do something better than anyone else, and before that skill/product turns into a commodity, is when those leaders get above-market returns. Any economist can tell you that -- and Agoric Enterprises hopes to demonstrate it.

Metric Object Publishing: Technology transfer

Metric Object Publishing PLC is a start-up launched by Metric Services, a 20-person software integrator focused on NeXT-based financial applications. Founder Job Maats, 40, is a Dutch-born, object-literate businessman who spent 14 years at Citibank and turned around Citibank Greece, one of its most profitable overseas operations by the time he had finished with it.
The company has eight people currently, and can also draw on the resources of Metric Group for the moment. They come to the business as users who would have liked such a service themselves, and more particularly as people in Europe who feel cut off from the rich information sea of the US -- in particular, California.

Although Maats has an e-mail address and the company in general spends a fair amount of time on the net, they don’t feel they have complete access to the best sources of object supply. (Surprise! A lot of people in the US feel the same way.) The company sees its natural supply base as the US, and its natural initial market as information-starved Europe, which the company knows reasonably well because of its past activities. For most US vendors of objects, Metric will offer incremental revenue. This month, Maats and his team went to Object World to round up suppliers; they had already been talking to the easier-to-find companies such as Borland, Sun and most of those listed in NeXT’s just-published catalogue.

The company should be able to establish a small, solid base in the European market; long-run, of course, it hopes to expand its product-acquisition office in Silicon Valley into a full-fledged sales and marketing subsidiary. By the time the net is truly global, and customers in Japan and Europe (let alone Russia and Brazil) are as well-informed as those in the US, Metric should be global as well.

The company’s initial plan is to publish a CD-ROM directory rather than fool around with online electronic delivery. Thus its initial funds (aside from VC money, which it is also seeking) will come from directory sales. Thereafter, it will pay the object authors royalties, with the percentage depending on conditions such as exclusivity, support requirements and the like. Won’t customers just look at Metric’s catalogue and then buy their objects direct? There are several possible answers to that: One, Metric may decide it can make more by selling information than by distributing objects; or two, Metric will sign some of its vendors to exclusives.

Third: Metric will have to be a true publisher/reseller, not a mere intermediary, and provide support. The company’s plan is to require the authors to provide objects and documentation of such quality that support is almost unnecessary. Metric will refer tough questions back to the authors, while Metric will provide consulting services for a fee. In practice, we think Metric may be a little optimistic about the support requirements for this kind of operation, but that’s a problem that should iron itself out.

The Class Library Inc.: Scarce information

Although it’s an incorporated company, The Class Library isn’t even a start-up yet; it’s an idea with a little bit of flesh and a ceo-to-be who still has a day job. This company’s initial idea is to license objects from horizontal software companies, but its goal is to be a vehicle for more cottage-industry vendors, says marketing vp Brian Cohen. It is working on establishing contacts with such companies as Borland, Lotus and Microsoft. The Class Library would provide an online browsing facility, and then would charge its customers to download the actual objects. TCL would take care of apportioning royalties to authors. It would also, says marketing vp-to-be Brian Cohen, currently a consultant, take care of quality control, relying
on some testing but primarily on bug and incompatibility reports from its customers, which would be maintained in a (probably object-oriented) database. Maybe the first ten customers for any new class library should be considered beta users and charged -- or compensated! -- accordingly.

Trilogy Development Group: Enabling technology

Of all the companies we know, Trilogy is the ideal one to manage the kind of information needed to run a successful object information service, whether online, on disk, or delivered by high-paid salespeople and consultants. That is, it has the necessary technology not just to keep track of objects, publish information about them and manage transactions, but also to model and manage the complex relationships among objects (inheritance schemes and other dependencies). Customers will purchase objects not just by attributes, but by the relationships and interactions of objects with each other. Yes, you can use an object-oriented database, but Trilogy’s configuration and modeling tools are what you need to build the database.

Trilogy’s position as a third-party manager of such information -- now from a variety of hardware vendors but ultimately from software vendors too -- gives it a unique edge in offering such a service, although it may sell its technology to other information providers as well. Think of the role Saabre plays not just in selling airlines tickets, but in supporting an army of value-added travel agents. And think how much more powerful it could be if it weren’t owned by American Airlines and restricted by antitrust issues.

We wrote about Trilogy last fall (see Release 1.0, 10-91). At the time, the company was a tiny start-up, with ten people operating out of one of those pre-fab Palo Alto offices. Now it has $4 million in venture capital from Greylock and New Enterprise Associates with a valuation of $20 million; 50 people; a referenceable installed base including HP, Silicon Graphics, LSI Logic and Pyramid; and a plausible goal of “65 percent of the hardware industry” as customers by the end of the year. Whether that’s installations in divisions of 13 of the top 20, or in divisions of vendors with 65 percent of the revenue, depends on whether IBM signs on. Clearly, the company’s biggest challenge is internal growth -- and keeping humble.

To recap, the product causing all the excitement is SalesBUILDER, an object-oriented configuration tool that maintains a knowledge base of a vendor’s products and components and how they all fit together. It solves the problem of configuring and pricing a product that makes sense; too often salespeople place orders that are underpriced, are missing vital parts (which must be supplied later for free) or simply can’t be built. The result is a lot of extra work in the factory, or worse yet fixing things for unhappy customers in the field. Instead, with SalesBUILDER, vendors can save a lot of money and keep a lot of customers happy. The system costs about $2500 a seat in a normal hardware company (50 to 1000 seats, typically). But wait, there’s more! For considerably less per seat, the vendor can give copies to his VARs; that both ensures they place valid orders and helps to win their loyalty. For even less, a pc vendor (for example) could give copies to large end-users who place frequent orders. It can of course keep track of customer configurations and suggest upgrades.

That’s the first phase, with each vendor buying SalesBUILDER and perhaps extending its use out into its channels, with runtime systems and dial-up ac-
ccess at reseller or customer sites. Each vendor will build and maintain its
own database of product information, including not just model and parts num-
bers but specific information about what works with or needs what else, how
the items can be combined (first fold along line A, then put tab B into slot
C....), and other information. Each vendor will also include a modicum of
information about relevant third-party products and software.

All that, of course, is a critical mass of information, kept from exploding
only by the fact that each "product object" (knowledge) base is kept sepa-
rate and managed by the vendor who owns it. Trilogy’s phase two, for early
next year, is NetBUILD², a single-vendor offering of a multi-vendor data-
base. That single vendor, of course, is Trilogy. This is where all the
information collected and maintained individually by Trilogy’s customers
combines into a valuable cross-vendor service available only from Trilogy.
Each vendor, of course, is eager to sell its products -- and thus eager to
supply information to Trilogy. They might wish to do it on their own, but
everyone by now has recognized that this is an open, alliance-led world.

The job of collecting and maintaining the information is daunting; HP, for
example, has hundreds of changes a day, says Trilogy. (Shades of the air-
line business again.) This is far more than Trilogy itself could maintain,
although it will have to manage the cross-impacts of other vendors’ products
that work with HP’s -- for example.

This very volume of work is mandates an automated way of representing and
maintaining the information, so that conflicts can be detected and in most
cases resolved electronically. Trilogy provides a unique service for now
both in being a way to handle these issues electronically, and in handling
them across vendors. If it doesn’t get lax or arrogant, it will have a vir-
tually unassailable market position for the foreseeable future.

THE SUPPLIERS

Like software vendors in general, virtually every object-oriented vendor is
trying to collect a coterie of supporting ISVs. Given the technology that
they’re using, they naturally get into the business of reselling or at least
promoting third-party class libraries. The primary forms this takes include
"partner programs," generally including catalogue listings and referrals.
For example, NeXT has just released its Objectware catalogue with more than
100 objects from more than 50 different vendors. The most frequent price,
however, is "call for information." After that, the most frequent is $99;
the most expensive is the Itasca 00DB for $3995 -- not a class library per
se, but a large server (an object, if you like).

Many vendors also offer bulletin boards where they can post notices and of-
fer support to their customers in turn. These bulletin boards (many of them
operated over CompuServe) are growing rapidly. They are certainly the
kernels of several of tomorrow’s electronic markets, even though those af-
filiated with a single vendor are suspect in some eyes.

--------

2 Maybe a better name would be Alliance Manager, for how it will enable
the industry’s alliance-mad vendors to handle all the products for other
vendors that they have agreed to co-market.
Separately, you can find listings for class libraries in the back of most object magazines. They have prices, restrictions on reuse (akin to runtime licenses or royalties required of developers buying databases and other tools). Many others are available over the net. How do customers know whether these will work with what they’ve got? They read the descriptions; they call and ask; they hope.

ParcPlace: Parcaged objects

ParcPlace built its business around Smalltalk; the company was a spin-off from Xerox of the famous Smalltalk team led by Adele Goldberg. Now the company is expanding vigorously into the C++ marketplace as well, but its catalogue of about 50 vendors still covers mostly Smalltalk services and products. The largest of these is Xerox’s Analyst, a large-scale framework for rich data analysis. Most of the objects are considerably smaller, says Goldberg: "Our customers, especially the corporations, tend to build their own proprietary application frameworks and base classes in-house." They go outside only for less mission-critical objects and tools and editors.

ParcPlace also offers ParcBench, a free dial-up electronic bulletin board that features offerings and news of the Smalltalk community. Goldberg has found that Smalltalk users are more prepared to pay for objects than C++ users, who tend to come from the university community and share code over the net.

Borland: Catalogue++

Borland has by far the largest installed base of C++ programmers through its Borland and Turbo C++ compilers, with about 650,000 copies sold. The Borland (professional) compiler comes in two versions -- for $495 without class libraries and $749 with classes and source code. (Borland would not break out more specific sales figures, except to say that the professional products sell more.)

Borland has a registry of about 150 third parties offering C libraries, including about 50 offering C++ class libraries. That information is available to any customer who calls to ask for libraries in a particular category. The volume of such calls is increasing, and so the company plans to publish a catalogue later this year that will include basic product information along with vendor contacts. The catalogue will be included as a booklet in all new boxes of C and C++. (And the NT version, which will be on CD-ROM, will include such a catalogue on CD-ROM, but that won’t be until NT ships.)

So far, most of the classes in the catalogue are fairly general or horizontal, for foundation objects, user interface, communications and the like, says David Intersimone, director of developer relations for Borland languages. "There are few application objects," he says. "They still seem afraid of giving away their technology." Although Borland doesn’t get involved in the commercial transactions, Intersimone notes that most of the vendors sell licenses to their class libraries outright. This is a little like selling a database tool with unlimited runtimes... But neither developers nor users really like the intricacies of keeping track of reuse.
Rogue Wave: Power under the surface

Rogue Wave is one of the major suppliers of foundation classes broadly defined; its Tools.h++ is probably the leading independent class of data structures. (Its View.h++ GUI classes are also doing well.) Several of the leading compiler vendors -- Liant, Metaware, Jensen Partners of the UK -- have standardized on its Tools.h++ and bundle it with their compilers (and paid a flat fee to do so). Microsoft, of course, does not, and Borland hedges on the question; it works closely with Rogue Wave and occasionally bundles some other Rogue Wave classes such as Math.h++ and Linpack.h++. CenterLine is also a big user and reseller of the Rogue Wave libraries with its ObjectCenter programming environment.

Rogue Wave was founded by Dr. Thomas Keffer of the University of Washington in 1989 to capitalize on the growing market for C++ classes. It has about 16 people overall and revenues of $1 to $2 million, the majority from the DOS market even at lower unit prices. He hopes that by providing the best classes cheaply for all vendors to resell, he can keep the market from coalescing around a single vendor. Of course, you could say it would then coalesce around Rogue Wave, but Keffer positions Rogue Wave as a small independent company with no vested interests -- unlike, say, HP or Sun. "We're so far down the hierarchy that no vendor will want to differentiate at that level," says Keffer. "We are non-controversial nuts and bolts. But before prefab walls and pre-hung windows can be constructed, we must settle on standard screws and wall sockets."

Of course, Rogue Wave has gained tremendous market exposure even though its revenues are small. Offering Tools.h++ so freely will now allow it to make money from its other libraries -- not so much because of technical dependence on Rogue Wave, but because of the visibility it has gained. Overall, the company has about 4000 to 10,000 fabricator customers, and countless other people who use its objects unknowingly in their applications.

Serius Corporation: Seriusly low pricing

Serius (see Release 1.0, 12-91) started out by allowing any applications developed with its Serius Developer ($595) and Programmer ($395) tools and classes to be resold for free, using the database compiler model. That got the company nicely established, to the point where it now has over 3000 customer tool installations. Now, however, it has rationalized its pricing to generate revenues from what it hopes will be a growing base of end-users of the objects that come with the tools, such as a spreadsheet/calculation object, a multimedia engine, or a database.

Its new pricing scheme will be much more along the current application model, where users are charged per-copy prices. The twist is that with object technology that's not per-application pricing. Several different applications may use the same object -- or one application may use several objects. "We really see these objects as being like fonts or inits or system accessories," says Serius founder Joe Firmage. "The user buys them for a small amount, puts them in his system, and uses them as needed. They're visible chunks of functionality; he can see them on his desktop as icons." The pricing will be on the order of $10 to $30 per object. A company that builds applications with the Serius tools can distribute them easily for use with each employee's own objects (possibly site-licensed).
ELECTRONIC MARKETS

A number of established electronic markets will be well-equipped to handle objects when demand for objects spreads beyond a small circle of friends.

CompuServe: The foundation market

CompuServe is already a marketplace for a variety of software components and software-related support and information, primarily through vendor-sponsored "forums." Generally, it makes its money from connect-time charges; it also remits a portion of those fees to the vendors (market-makers?) who run the various forums. Many of the vendor-sponsored software forums offer the distribution of software and objects either formally or informally as part of paid or unpaid customer support. How the vendors charge their customers for this support is the vendors' own decision. However, when there are vendor charges, customers can pay through their CompuServe accounts; CompuServe handles the billing for a small service charge, typically around 10 percent. The power in this medium remains more or less with the vendors, for now, although CompuServe is finding it an increasingly important source of revenues. CompuServe exercises no editorial control and does not select or feature products.

Separately, CompuServe has two distinct transaction-oriented software markets: Softex, in existence for over five years, and Shareware Registration, which is just now being launched. Softex is a small business, with a few hundred offerings. It hasn't been promoted heavily, but interest is growing, and CompuServe may start to put more resources into it. Buyers pay through their monthly CompuServe bills, and vendors receive their royalties after CompuServe takes its cut -- typically 40 percent. This gives the author 60 percent, an unusually high royalty, but the products tend to be low-priced and there are no packaging/publishing costs. Support and marketing are also generally minimal.

Sensing the interest in Softex, CompuServe recently started its Shareware Registration service. Like Softex, it's more a convenience than a real marketing vehicle. CompuServe's Softex and other areas on CompuServe can download the software directly; Shareware Registration customers typically acquire it on another CompuServe service or elsewhere and simply use CompuServe to register it. Both services are especially useful to customers living abroad, who avoid the hassle of trying to order through 800 numbers that aren't toll-free outside the US, currency conversion and bank transfer complications and fees, and time-zone disparities. Although currently CompuServe is more active in actual software distribution than its vendor customers, in the long run we expect to see it become a bigger part of the support forums. Eventually foundation class vendors will probably sponsor/manage third-party markets for dependent class libraries and assurances of support and quality.

UniLinx: UNIted market

UNIX System Labs recently announced the formation of UniLinx, a global network service to provide product information and technical support for the UNIX market. Again, although it's not formally directed at the distribution of objects, it will provide a platform for such activity. The concept was
dreamed up separately by Roel Pieper, CEO of USL, and by Peter Watts, CEO of Tangent International of the US, who also brought in Fujitsu. Their exact investments have not been disclosed, but all three have "enough skin in the game" to be committed to making it work, says Watts. Fujitsu gives the network a global reach and will be providing the technical capabilities -- networking and data/object-management software and ISDN capabilities. USL gives the venture a focus -- UNIX software and services -- and visibility. And Tangent, in the form of Watts, 39, will give it strategy, management and market expertise.

Watts is uniquely suited to the task of UniLinx CEO: He's international (raised in London), he has experience in the online database business, and this is his fourth start-up. He co-founded Tangent in the UK in 1974 to be a mainframe-oriented consulting house; he started its US operations, focused on UNIX, in 1979; and he left Tangent to found kiNexus, a student-recruitment database service, in 1987. kiNexus now has the resumes of 140,000 students online, and AT&T is one of its major customers. Watts got to know UNIX System Labs in particular, however, when he returned to Tangent in the US. One of the company's major moves was to become a specialist in Tuxedo, USL's transaction-processing software (and a strategic item in turning UNIX into a commercial offering). Each hardware vendor he talked to had the same problem -- getting information and technical support to its customers. Hence the idea for UniLinx.

USL, meanwhile, needed a way to rally support for UNIX, not just from customers, but from the people who really count in making a sale -- developers, value-added resellers, ISVs and consultants. Moreover, much of UNIX's stronghold is overseas, where support has been the poorest.

A joint venture avoids looking too heavy-handed in a market still sensitive to AT&T's size (even though its power is waning). Obvious suppliers of information include AT&T's own NCR computer unit, which has pretty much abandoned its proprietary products in favor of UNIX and Microsoft's DOS and Windows; Univel, a joint venture between Novell and USL; and a host of second-tier UNIX companies. Less likely to join in initially are IBM and DEC, even though they too sell UNIX systems.

UniLinx will basically be a medium for anyone selling UNIX products or services to promote and support those items. It should support not just textual information or code, but also demos, online conferencing for sales or interactive support, and software delivery. UniLinx will charge vendors a small storage fee to maintain and store their information, and transaction fees for various kinds of activities from customer contact to actual sales and software delivery. The precise details, of course, haven't been worked out and will probably be determined by negotiations with each vendor -- at least until a few standard arrangements emerge. (Another class with potential for inheritance and subclassing!)

Thus UniLinx would probably be most intensively used by intermediaries who would sell their services to customers. The system itself won't have editorial input from UniLinx, which will basically handle information from all comers. Its contribution will be limited to providing a telecommunications platform and search/data management software -- although of course that ends up having a fairly strong influence on what content can be represented and interpreted. Each vendor will provide its own information -- some directly
to its own user/prospect groups, or for more general consumption. Unlike Trilogy, UniLinx won't really attempt much integration of the information from different sources. On the other hand, its service would be a perfect platform for vendors such as Metric or TCL or Trilogy, or for Arthur Andersen to broaden its reach -- or for the mythical Object Ratings Service. On the other hand, a low-service reseller of objects probably wouldn't find it very interesting, since UniLinx and its brethren are trying to make the delivery (as opposed to the content) of information into a commodity service. Separately, Watts sees Tangent as a beneficiary (along with many other firms) of UniLinx in support of its UNIX consulting business.

The hitch with UniLinx, of course, is that it won't give its customers a competitive edge over each other -- just over non-UNIX computer resellers.

ZiffNet: Publisher's paradise

ZiffNet is part of Ziff's Desktop Information Services, which also includes Computer Library (with CD ROM products such as Computer Select and Support on Site). ZiffNet is an information and product selection and advice service for which users pay a $2.50 monthly subscription fee plus time charges; they don't, for now, pay for information or software specifically. ZiffNet includes downloadable software (free or shareware), plus a variety of editorial information and advertising mostly culled from the Ziff print publications and buyers' guides. Thus ZiffNet is a market intermediary, selling information about goods rather than the goods themselves.

One big provider of shareware over ZiffNet is Public Brand Software, a shareware distribution house recently acquired by Ziff. Yes, shareware may be acquired for free, but the business of distributing it -- in PBS's case, testing, selecting, rating and promoting good products -- is an information service for which customers are willing to pay by buying their disks from PBS's catalogue -- or by paying the time charges to download the software from ZiffNet. (Ziff gets a cut of the time charges from CompuServe.) The availability of PBS on ZiffNet is one of ZiffNet's appeals to ZiffNet customers. And PBS's "editorial engine" was its allure to Ziff.

Separately, Ziff has just launched Ziff Buyers' Market, an advertiser-funded (no time charges) online information service which provides product information and vendors' phone and fax numbers. Fundamentally, the goal is to get customers to call the vendors or resellers directly rather than purchase through the system. ZBM is distinctly different from Ziff Buyers' Digest, which provides unbiased product evaluations performed by Ziff for which the user pays $14.95 per product area. In ZBM, the advertisers pay to get their message across, whereas in ZBD users pay for Ziff's independent research.

While there's no reason not to offer a transaction service either on ZiffNet or ZBM, says vp online services Dave Shnaider, the company hasn't gotten around to it yet. There's also the little matter of channel conflict: While vendors would be delighted with a new channel, the reseller community which uses Ziff heavily for advertising might not be pleased at the new competition. Overall, ZiffNet has about 100,000 regular users. ZBM gets about 3000 to 4000 sessions each week, but that number is still growing fast.

ZiffNet generally operates over CompuServe; you can use it directly through CompuServe without being a CompuServe subscriber. It is also newly available on Prodigy.

*Release 1.0*  
32 July 1992
FOR FURTHER READING

Lavoie, Don; William Tulloh and Howard Baetjer, "Order in Complex Systems: object-oriented programming and the economists' critique of central plan- ning," 1991. Available from the authors. This is an intriguing paper but (surprisingly) too short; it leaves some interesting questions unaddressed -- for example, how markets might encourage specialization of objects.

"The artificial economy project," a research proposal available from the Santa Fe Institute, (505) 984-8800. This describes an existing prototype artificial economy and a proposed object-oriented model to be used in modeling complex processes. In explaining the planned use of OOPS, the report says (among other reasons) that the system must be easily extensible, "to adapt it to many different concrete problems, involving different institutional structures and modes of decision-making by individual agents... Institutes and behavioral styles at the micro level matter: Their com- plexity, in terms of the feedback and energy flows they induce among the various agents, determines the potentiality of the system for emergent orga- nization." That is, you can't predict what will happen by using equilibrium theory, macroeconomics and financial aggregates; consider SimCity.

Cox, Brad, "What if there is a silver bullet...and the competition gets it first?" Journal of Object-Oriented Programming, June 1992.

Ryoichi Mori and M. Kawahara, "Superdistribution: the concept and the ar- chitecture," Transactions of the IEICE, 1990. This actually deals more with semiconductors than with software, but starts to with the concept of using a chip to meter the use electronic information. "Superdistribution" is analogous to superconductivity, where electricity flows freely without resistance.

***

COMIC RELEASE 1.0: NO, WE WON'T TELL!

We recently received the following e-mail message from an e-mail tool vendor, shown in its original form except for the header and the black-outs:

To: ESTHER
From: [REDACTED]
Subject: [REDACTED] Non-disclosure

EstheHopefully, you have received the information about [REDACTED] and our products. I'd also like to send you details on our marketing strategy and direction, but of course I need assurance that you'll hold this information in strict confidence.

[REDACTED] uploads all aaaaages via an error correcting protocol, so I guess the challenge with the last

Release 1.0

32 July 1992
RESOURCES & PHONE NUMBERS

Howard Baetjer, Agoric Enterprises, (703) 250-4760; fax, (703) 250-3532
Gayle Pergamit, Chris Peterson, American Information Exchange, (415) 903-1000, 903-1070 or 903-1051 or 903-1065; fax, (415) 903-1093
Eric Herr, Carol Bartz, Autodesk, (415) 332-2344 x4884 fax, (415) 289-4719
David Intersimone, Borland, (408) 439-4805; fax, (408) 439-9343
Brian Cohen, The Class Library, (404) 395-6448; fax, (404) 315-0293
Barry Berkov, CompuServe, (614) 457-8600; fax, (614) 457-0504
Peter Sprague, Jim Kolleger, Cryptologics, (212) 755-3282; fax, (212) 755-3436
Robin Vasan, Infinity International Financial Technology, (415) 964-7800; fax, (415) 964-9844
Sat Tara Khalsa, InfoNow, (303) 442-6666; fax, (303) 786-8473
Brad Cox, Information Age Consulting, (203) 868-9182; fax, (203) 868-0780; bradcox@aol.com or bradcox@infoage.com
Job Maats, Metric Object Publishing PLC (London), 44 (71) 495-4400; fax, 44 (71) 499-8334; job.metric@asmec.co.uk
Adele Goldberg, ParcPlace Systems, (408) 773-7406
Tom Kaffer, Matt Steinauer, Rogue Wave, (503) 754-2311; fax, (503) 757-6650
Joe Firmage, Serius Corporation, (801) 261-7900; fax, (801) 261-7910
John Price, Joe Llemandt, Bill Read, Trilogy Development, (512) 794-3822 or (415) 321-5900 x110; fax, (415) 321-8900
Peter Watts, UniLinx/Tangent, (212) 809-8200; fax, (212) 509-6578
Ed Chatlos, Roel Pieper, UniLinx/UNIX System Labs, (908) 522-6516/6170
Dave Shnaider, ZiffNet, (617) 252-5220; fax, (617) 252-5551

COMING SOON

- Groupware tools.
- Parallel computing.
- Commercialization of the internet.
- Infrastructure for groupware (August).
- Pen stuff.
- Constraint-based reasoning.
- Text tools.
- And much more... (If you know of any good examples of the categories listed above, please let us know.)

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. A companion publication, Rel-EAST, covers emerging technology markets in Central Europe and the Soviet units. Editor: Esther Dyson; publisher: Daphne Kis; circulation & fulfillment manager: Robyn Sturm; executive secretary: Denise DuBois; editorial & marketing communications consultant: William M. Kutik. Copyright 1992, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we gladly arrange for reprints or bulk purchases. Subscriptions cost $495 per year, $575 overseas.

Release 1.0 32 July 1992
## Release 1.0 Calendar

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Location</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 13-16</td>
<td>The 9th International Development Center conference - San Diego.</td>
<td>San Diego</td>
<td>Practitioners tell war stories and vendors offer advice. Sponsored by the Development Center Institute. Call Keith Cox, (317) 846-2753.</td>
</tr>
<tr>
<td>September 13-16</td>
<td>The conference on practical leading-edge computer networking</td>
<td>Minneapolis</td>
<td>Sponsored by IEEE. Call Steve Bell, (415) 966-7926.</td>
</tr>
<tr>
<td>September 14-16</td>
<td>DataStorage92 - San Jose. Sponsored by Disk/Trend.</td>
<td>San Jose</td>
<td>Call Darlene Plamondon, (415) 508-0118.</td>
</tr>
<tr>
<td>September 14-18</td>
<td>Comdex/Sucesu-SP '92 - Sao Paulo, Brazil. South America's largest computer trade show and conference. Sponsored by The Interface Group.</td>
<td>Sao Paulo</td>
<td>Call Susan Hicks, (617) 449-6600.</td>
</tr>
</tbody>
</table>

Release 1.0 31 July 1992
<table>
<thead>
<tr>
<th>Date Range</th>
<th>Event Description</th>
<th>Sponsor</th>
<th>Contact Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 22-24</td>
<td>*UNIX EXPO - New York City. Sponsored by Bruno Blenheim. Call Annie Scully, (201) 346-1400 or (800) 829-3976.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 22-25</td>
<td>Seybold San Francisco - San Francisco. Sponsored by Seybold Seminars. Call Kevin Howard or Beth Sadler, (310) 457-5850.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 23-25</td>
<td>Virtual Reality '92 - San Jose. Sponsored by Meckler. Call Gloria Allen, (203) 454-5840 or (800) 635-5537.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>October 11-14</td>
<td>*Third annual East-West High-Tech Forum - Prague. Sponsored by EDventure Holdings. Contact: Daphne Kis, 1 (212) 758-3434; fax, 1 (212) 832-1720; e-mail, MGI 511-3763 or Computerve 75140,761.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>October 18-22</td>
<td>OOPSIA '92 - Vancouver. Sponsored by ACM. Call Rebecca Wirff-Bock, (503) 242-0725 or John Richards, (914) 784-7731.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>October 20</td>
<td>New York PC user group meeting - New York City. Sponsored by NYPC. With Andy Grove. Call Jim McMullen, (914) 245-2734.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>October 27-29</td>
<td>PC EXPO - Chicago. Sponsored by Bruno Blenheim. Call Annie Scully, (201) 346-1400 or (800) 829-3976.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Event</td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>November 9-12</td>
<td>Software maintenance '92 - Orlando. Sponsored by ACM. Call Vaclav Rajlich, (313) 577-5423.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>November 16-20</td>
<td>*Comdex - Las Vegas. The biggest US show of all. Possibly featuring a Soviet pavilion this year. Contact: Peter Young at Interface Group, (617) 449-6600; fax, (617) 449-6953.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nov 30-Dec 4</td>
<td>Hypertext '92 - Milan, Italy. Sponsored by ACM. Contact: Julie Eitzer, (212) 626-0604; Paolo Paolini, 39 (2) 2399-3520; e-mail, <a href="mailto:paolini@ipmell.polimi.it">paolini@ipmell.polimi.it</a>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>December 8-10</td>
<td>DEXPO - Anaheim. Sponsored by Miller Freeman. Call KoAnn Tingley, (617) 232-3976.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>February 21-24</td>
<td>**EDventure Holdings PC (Platforms for Computing) Forum - Phoenix. You read the newsletter; come meet the players and try their tools. Call Daphne Kis, (212) 758-3434.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>March 24-31</td>
<td>CeBIT '93 - Hannover, Germany. Sponsor: Hannover Fairs USA. Everything you’re interested in, and a lot more. Call Mette Fisker Peterson, (609) 987-1202; Geide Schlichting, 49 (511) 89-0; fax, 49 (511) 89-3 2626.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mar 31-April 3</td>
<td>Seybold Boston - Boston. Sponsored by Seybold Seminars. Call Beth Sadler or Kevin Howard, (310) 457-5850.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>April 24-29</td>
<td>INTERCHI '93: Human factors in computing systems - Amsterdam. Sponsored by ACM. Call Carol Klyver, (415) 738-1200.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Release 1.0
31 July 1992
SUBSCRIPTION FORM

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 must be enclosed. Multiple-copy rates available on request. Satisfaction guaranteed or your money back.

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Company</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Telephone

☐ Check enclosed.

☐ Charge my
  ☐ American Express  ☐ MasterCard  ☐ Visa

Card Number ___________________________ Expiration Date ___________________________

Name on Card ___________________________ Signature ___________________________

☐ Please send me information on your multiple-copy rate.

Please fill in the information above and send to:

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

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

Daphne Kis
Publisher