



Explorers and expanders, both early and late

SHANE GREENSTEIN

greenstein@kellogg.northwestern.edu

..... Explorers and expanders keep the engine of development running. They have inhabited commercial technology markets as long as such markets have existed. Although it is possible to explain these roles in abstract terms, it is easier to simply illustrate them based on recent events.

Early explorers and expanders

Explorers test a new value proposition or apply a technology in a new way. For example, intrepid vendors make few sales, change designs from one launch to another, and take large risks. Typically there is no cookbook for this activity. Explorers learn through their own experience *and* from the experience of others. They learn about, for example, the true scope of business opportunity or what a buyer values in a new design or service.

The Internet attracted a tremendous number of explorers. After privatization in 1992, the first explorers were IBM and MCI, companies that had provided the government contracts for Internet service. Other explorers were entrepreneurial spin-offs from the government service, such as PSINet and BBN Technologies, among others, as well as equipment firms, such as Cisco, which at the time did a modest business at universities.

A second set of early commercial explorers soon arrived, taking advantage of the creation of standards associated with the World Wide Web and the easy-to-use browser, both 1993 inventions. This

included Netscape, a late-1994 entrant. A few of the explorers still remain from the class of 1995, such as Yahoo, Amazon, and eBay, as well as too many basic Web building tools to list.

The hype of 1995 generated several more waves of explorers from 1996 to 2000, mostly dot-coms and telecom entrants. It is these firms, especially the post-1999 entrants, who captured the public attention with their excesses, such as the sock puppet at Petsmart.com. The public perception is a bit unfair, however. The majority of these firms were serious ventures started by venture capitalists who had no intention of losing money. Most of these firms had aspirations to be expanders but never got close to realizing their ambitions.

What is an expander? Expanders develop a low-risk value proposition for mainstream customers, those who want reliability, and shied away from most explorers. Expanders excel at organizational routines that scale into large businesses. Channels develop routines, standard designs dominate sales, and operational habits emerge, supporting mass market business. Such firms employ hordes of lawyers, accountants, and marketing experts who excel at smoothly developing business activity.

Successful expanders tend to be big. These types of businesses need high revenues to cover large overhead expenses. For example, most dot-coms failed because they had the overhead, but not the revenue.

Expanders typically follow explorers because expanders can only succeed by learning from the experiences of explorers. Moreover, explorers tend to stop arriving once the expanders succeed.

But the Internet is not like most markets. First, as noted, many potential expanders arrived too soon. Second, many *early* expanders led to *later* explorers, which led to even *more* expanders. This requires explanation.

Earliest expanders

IBM was the largest expander in computing markets prior to the Internet. Indeed, in 1994 IBM did act as an expander for the Internet access business that served large businesses, its primary legacy customers. However, for several reasons, IBM never moved its services into the home market.

Instead, AT&T acted as an expander of the access market for homes when it began offering WorldNet service to homes for \$20 a month in late 1995. It beat AOL to the punch by quite a bit, gaining a million customers in a few months.

There are a couple lessons in AT&T's bitersweet experience, so let's go a bit deeper. AOL made the most money in the home market, taking the expander mantle from AT&T. Why was that?

Back at the beginning many early explorers—AOL, CompuServe, and plenty of others—thought users would tolerate per-minute charges to use information services. Bulletin boards and cell phone

pricing worked this way because it saved on the use of the costliest scarce resource, the network access point.

AT&T, in contrast, introduced flat-rate pricing. It bet it would attract new users. It bet that the costs would not become so large if it used its extra-large modem banks efficiently. AT&T turned out to be right, but only partly so. This feature attracted *some* new users to AT&T, at least at first, but it was not enough to make most AOL users switch.

Then the strategy ran out of steam. AOL matched the flat-rate pricing within a year. After AOL got its flat-rate pricing to work well, AT&T had no key differentiator. A year later, the next set of new users wanted flat-rate pricing *and* varieties of content, which AOL had figured out and AT&T had not. In addition, by 1997 other Internet service providers had entered the market with many different types of access business models, and each one of them took a small piece of the growing pie.

There are three immediate lessons in this experience. First, as expected, competition among expanders took a while. Expansion did not cease after one big expander had partially succeeded.

Second, usually a mix of organizational activities and technical capabilities leads to long-term commercial success as an expander. For example, AOL offered superior marketing, Web page design, Web community building, and tailored content offerings, not just branding or technological prowess.

Third and more deeply, appealing to mass market users differs from appealing to early users. Often most early users seek a technological utopia and pay for it. Mass market users, in contrast, avoid expensive, but perfect, designs if they can get a very cheap, yet reliable, one. In other words, great technology usually does not determine competitive outcomes between expanders; nontechnical features often do.

Late explorer

There is one other surprise. The description I just laid out did not end the story. The last half decade has given rise to multiple late explorers.

Late explorers take advantage of all the

expansion during the first generation. That is a double-edged sword. A late explorer *builds* on the past, but it does business in an environment *constrained* by the compromises of the successful expanders.

Because late explorers walk into known market territory, they face heightened competitive tensions. Late explorers do not need to appeal to technically sophisticated users, as did the early explorers. Instead, late explorers can look for a radically different value proposition than that of previous expanders. Google is a good example. It took an established product category—online search—and redefined the activity. Then it deftly built on that entrance.

Late exploring also has little margin for error. For example, another successful late entrant was NetZero, which entered the market in early 1999. It had a free-ISP model, which it eventually merged with Juno. It has built volume and maintained a consistent managerial vision. This focus and discipline has kept NetZero from flaming out like so many others.

Late explorers also face issues about the *timing* of their action. For example, this generation of broadband firms sells to a home Internet user, someone who knows what to do with broadband. That means these late explorers do not need to do nearly as much handholding as AOL. In addition, most of the cable firms and DSL (digital subscriber line) vendors have unique assets—lines in the right locations—that others cannot easily replicate. That explains why these companies could take their time and not suffer competitive consequences. In other words, from a business point of view the seemingly leisurely pace of the broadband firms is probably good strategy.

This brings up a key point. If they can, late explorers might be better off deliberately *waiting* for the early builders of scale to finish competing. Why? Practically speaking, you must live through a period of exploration and expansion to understand why it turns out the way it does. Sometimes, it is not worthwhile to take action—such as building a business on top of the mass market—until market participants have learned the appropriate lessons about what sort of compromises work.

On the other hand, few firms ever have the luxury of waiting. This explains why so many of the serious dot-coms entered sooner than was wise. They anticipated a role that they did not (and could not) learn how to do soon enough, but competitive pressures induced many to gamble with their timing.

The whole is greater than the sum

The most fascinating late explorers are those whose exploration would not be possible in anything less than a nearly ubiquitous network. These firms must enter later because their business depends on size. Size creates opportunities that did not previously exist.

Here is one example of many: The Wi-Fi market (that is, IEEE 802.11b/g) initially attracted many explorers. Just look at any hot-spot map for a major city today. It is possible to surf on your laptop at many locations other than the airport.

Many firms' actions made that happen. Intel's blunt entry into the Wi-Fi market with its Centrino brand had an effect, but Intel was not alone. The improvements by battery makers were very important. So too were the decisions by Starbucks and McDonald's, not to mention thousands of local Internet cafes and their offerings of cappuccino. Suppliers such as Dell, Best Buy, and Circuit City, to name a few, also contributed. It is time for expanders.

You can see similar patterns at various stages all over much of data communications today. It is possible to write extensively about the accumulation of explorers (and even some expanders) associated with application service providers, Bluetooth applications, voice over Internet Protocol (IP), flavors of peer-to-peer services, WiMAX, various types of virtual private networks, Extensible Markup Language appliances, or IP videophones.

Moreover, a common headline today heralds the fact that e-commerce is starting to turn a profit, unlike prior attempts, which were before their times. That is just another way of saying that the new media has figured out that too much exploration came too soon.

Overall, there is something existentially

unsettling about this pattern of explorer, expander, followed by another later explorer. The sum total builds to something bigger than any firm could do by itself.

More to the point, this collective learning is not organized around any collective purpose. Each firm benefits from the experience of the other, but no firm deliberately tries to fail so everyone else can learn the right lesson. Things always move forward, but the lessons arise haphazardly. So too does the accumulation of learning.

Where does it lead?

The development of the railroads had a similar dynamic pattern 150 years ago. Then, too, the business and technology of railroads evolved together.

The first tracks were built from iron. It was years before the engines were reliable. Then along came steel rails, which set off another round of engine and car redesign. It went on for decades until the modern automobile arrived.

Are we in for a run of that length too?

Perhaps we are. Today's cycle is outside of anyone's control. It builds in fits and starts, inviting more explorers and more expanders. Only a fool would predict where it is going or when it will stop.

For further information on this or any other computing topic, visit our Digital Library at <http://computer.org/publications/dlib/>.

Micro Review

continued from p. 80

mercial vendors began to diverge from the original model. Much of this divergence became embodied in the SQL language. As an aside for those who care about such things, the name SQL originally stood for *Structured Query Language*, and many practitioners pronounced it "sequel." In its standardized versions, however, the name does not stand for anything. It is just the three letters, S-Q-L, and that is the way Date pronounces it, though he acknowledges that the other pronunciation is still widespread.

The problems with SQL have little to do with its name. Problems arise from its abandonment of fundamental principles. For example, SQL fails to keep the data model separate from its implementation. Its basic structures are tables, which differ from relations in that they have ordered rows and columns, and can have duplicate rows. They differ in other ways too, but you'll have to read Date's book to find out.

Date prefers to use the language Tutorial D, which he and Darwen define in the *Third Manifesto* book. Date believes that the original relational model does not need to—in fact, should not—change to accommodate object-oriented features. The original model deals with relations on domains. Domains are arbitrary sets, and a relation on domains T_1, \dots, T_n is approximately a set of sets, $\{t_1, \dots, t_n\}$, plus a mechanism (based on a heading and annotations in the body)

that allows elements to maintain their identities as members of their respective domain sets. A theory of types and inheritance adds structure to the set of domains. Date regards this structure as orthogonal to the original relational model, because the original model assumes nothing about the domains. The manifesto uses the term D to represent any language that provides the orthogonal object-oriented extensions called for by the manifesto. Tutorial D is a specific example of D. That means that Tutorial D (unlike SQL) is relational and provides object-oriented features in a way that is orthogonal to the relational model.

This book uses Tutorial D in places, but it does not go into detail about the theory of types. It focuses on providing a brief, but not terse, presentation of the original relational model, contrasting it with current commercial products and discussing contentious issues such as the wisdom of allowing null attributes. It contains many thought provoking exercises. An O'Reilly Web site provides answers and further discussion. One drawback of the book is its poorly designed index. O'Reilly should invest in a good one for the next printing.

Date complains about the product-oriented nature of computer science instruction, which leads to a number of widespread misconceptions about the relational model. He hopes by his presentation, which relies on principles rather than the quirks of SQL or specific products, to correct these misconceptions and produce a greater understanding of and

appreciation for the relational model.

To help you decide whether you need to read this book, Date suggests that you try to answer the following questions:

- What exactly is first normal form?
- What's the connection between relations and predicates?
- What's semantic optimization?
- What's a join dependency?
- Why is semidifference important?
- Why doesn't deferred integrity checking make sense?
- What's a relation variable?
- What's nonloss decomposition?
- Can a relation have an attribute whose values are relations?
- What's the difference between SQL and the relational model?
- Why is The Information Principle important?
- How does XML fit with the relational model?

Date does not specify the number of questions you must answer correctly to be exempt from reading the book. I suspect that many experienced practitioners will fall far short of 100 percent. If you are in that category, or if you just want a clear account of the relational model and its underlying principles, you should read this book.

For further information on this or any other computing topic, visit our Digital Library at <http://www.computer.org/publications/dlib/>.