



Is Your Middleware Dead?

Steve Vinoski • IONA Technologies

Your company has an application-integration problem, and you must find a solution. You're pretty sure you need some kind of integration middleware, but the numerous approaches and vendors from which to choose don't make your decision easy. How do you choose the one that will best solve your problem? Perhaps more importantly, how do you know if the middleware you choose will be economically viable for as long as you plan to use it? What happens if you choose something that – as far as “the market” is concerned – is about to fade away and die?

On the surface, these questions aren't unusual or unreasonable, but the answers that our industry usually provides tend to focus on technical pros and cons. Thus, the answers are often way off base because determining any solution's applicability to a particular problem must include analysis and understanding beyond technical merits. There are several nontechnical issues to consider, foremost of which are the relevant market effects. Several recent articles and discussions reminded me of just how much confusion and inaccuracy permeates most technical discussions that involve market issues. So, for this installment, I want to discuss technology markets, not technologies.

RIP Middleware

In early 2004, Jonathan Schwartz, executive vice president of Sun Microsystems' Software Group, predicted “the end of middleware.”¹ One fatal flaw with his prediction is that it's predicated on the unlikely universal adoption of the Java Enterprise System (www.sun.com/software/javaenterprise/system/). In other words, if everyone just adopted the Java Enterprise System, then middleware integration would become completely unnecessary. Ironically, this is because the Java Enterprise System *is* middleware.

Some threads from the July and August 2004

Usenet comp.object.corba newsgroup also revealed that technologists don't necessarily understand markets. One of these threads – “Is Corba dying?” – began with an observation that activity and traffic in the newsgroup had decreased, and wondered whether that was an indication that Corba was near death. A long-winded discussion involving more than 50 postings ensued.

Some postings stated that J2EE has taken over Corba's leadership, while others complained that J2EE is a Java-only solution. Still others bemoaned their belief that Corba 3.0, with its new component model, still isn't getting the attention it deserves despite being technically more powerful and advanced than either .NET or J2EE. A few postings did venture into nontechnical territory, but most offered outlandish theories, such as those claiming that Corba vendors had charged too much for their products and that their greed had ultimately limited market size. Overall, most postings limited their arguments to Corba's technical merits and approach viability, expressing dismay at newer technologies that appeared to just reinvent the wheel – poorly, of course.

Another thread – “Contribution of SOAP” – in the same newsgroup asked for opinions regarding SOAP's main technical contribution. The thread took a similar tack, with several postings opining that SOAP wasn't only technically inferior to Corba, but was largely just a reinvention of the wheel and, thus, had made no real technical contribution. (I argued in this thread that for many projects, SOAP reduced the size and complexity of the integration infrastructure required on the endpoints, but nobody responded to my posting.)

Schwartz's end-of-middleware article and the newsgroup threads share a common theme: fear of middleware's death. The article preys on such fears, essentially saying, “We're experts, and we say that the middleware train is coming to a stop.

Are you just going to ignore it and sit here while your competition leaves you in the dust?" On the other hand, the newsgroup discussions express a fear of the unknown, wondering how and why a clearly superior technology could fail to be universally accepted and acclaimed.

Technology-Adoption Life Cycle

Schwartz's ominous prophecy, like other similar predictions before it and still others that will surely follow, isn't only idealistic and obvious, it's also utterly unrealistic. Chief among its shortcomings is that it completely misses the fact that different integration-technology consumers have different wants and needs that no single product or approach can fulfill. We technologists willingly accept that we don't all drive the same type of car, live in the same type of home, or eat the same foods. Why, then, do we insist on believing that there can be only a single correct integration-technology answer?

The Consumer Bell Curve

What's interesting about the newsgroup threads is that, except for a few postings, they are almost completely devoid of perspective beyond the Corba technology realm. For example, perhaps traffic in the comp.object.corba newsgroup had dropped because June, July, and August are summer vacation months in the US and Europe. Alternatively, the drop could be due to the rising popularity of other discussion forums such as weblogs and Web site communities like www.theserver side.com. But the primary reason could be that the primary consumer type that Corba currently appeals to might not be the type that posts to public forums and newsgroups.

Obviously, consumers are different, and their differences result in different technology-adoption decisions. A technology that is "too old" or "near death" for one consumer might be quite

appealing — perhaps even appearing new — to another. Fortunately, we already have a good handle on how different types of consumers adopt technology, thanks to the work of Geoffrey Moore and others.

In *Crossing the Chasm*² and *Inside the Tornado*,³ Moore provides thorough and compelling discussions about technology-adoption life cycles and how businesses can exploit them. Typically, a technology-adoption life cycle is a bell curve; the horizontal axis indicates time and

money to spend. These people believe that if a new technology can save time, money, or effort, exploiting it before anyone else gives them a competitive advantage. Because they bring money to the table that can fund development, early adopters are key to transitioning technologies from idea to pragmatic solution. Because they provide funding, they also typically demand numerous changes and new features. However, their control of the purse strings can sometimes be detrimental because their demands can

Obviously, consumers are different, and their differences result in different technology-adoption decisions.

the vertical axis indicates technology-adoption rate. The curve divides into five regions based on different consumer types:

- innovators,
- early adopters,
- early majority adopters,
- late majority adopters, and
- skeptics.

Leftmost on the curve are the innovators, technology enthusiasts who love to play with the newest approaches. To those outside the computer industry, all technologists appear to fall into this category, but we don't. In any group of technologists there's always one who's investigating brand-new approaches, playing with new gadgets, or learning a new programming language. Innovators rarely buy technologies, but they profoundly influence technology market development because they're first on the adoption curve. If they don't give a particular technology their nod of approval, it will end up going nowhere.

Next come the early adopters, consumers similar to innovators, but with

inhibit the technology's future acceptance in the mainstream market.

If a technology succeeds in its early market with innovators and early adopters, it then enters the mainstream market via early majority adopters. Between the early market and the mainstream market is the *chasm*, a region in which technologies can die by failing to find a foothold in the mainstream. Because early majority adopters are more risk-averse than those in the early market, such footholds fail to appear when technologies can't overcome the perception of being too risky for mainstream consumption. Early majority adopters need convincing proof that a new technology can help them gracefully evolve their systems to a newer, better place. Otherwise, they won't adopt it. The ability to evolve their systems is very important, as they want to keep what works rather than having to rip it out and replace it. The mainstream market phase is critical because it's where a technology's adoption rate is highest, which typically means it's where the most profit will come from.

Late majority adopters, sometimes called laggards, are even more conservative. For them, a technology has to work flawlessly, right out of the box, before they will buy into it – and it better be priced just right as well. Late majority adopters don't really believe that new technologies can help them, and they adopt them only when there isn't much choice.

At the bell curve's far right are the skeptics. They're so conservative that they don't actually buy new technologies directly. Rather, they buy technol-

would be well into something newer on the bleeding edge.

Similarly, participants in the comp.object.corba discussions about Corba's lifespan and SOAP's technical contribution largely failed to mention any effects from the technology-adoption life cycle. Only one posting (other than my own), written by my friend and longtime middleware researcher Douglas C. Schmidt of Vanderbilt University, correctly pointed out that Corba, which has already been in use for more than a

How do you choose the middleware that will best solve your problem?

ogy as part of other products in which it's so deeply embedded that, for all practical purposes, it's invisible.

Based on these categories, it's not hard to see why a technology that an early adopter considers to be old and boring could seem new and risky to a laggard. In the integration-technology context, these different consumer categories mean that not only is it impossible to have a one-size-fits-all solution, it's also meaningless to try to hold a general discussion about whether a particular technology is dying or dead without also considering the target consumer category.

Practical Effects

In the context of Schwartz's dire prediction, is the Java Enterprise System capable of appealing to all five consumer categories and, thus, truly bringing about the end of middleware? A broad appeal is very unlikely. If a technology were to appeal to early adopters, majority adopters would probably view it as too risky. If it were to appeal to majority adopters, it would probably be old hat to early adopters but not yet ready for skeptics. And by the time skeptics adopted it, early adopters

decade, was naturally moving along the technology-adoption life cycle curve toward the conservative end. This move explains why Corba has become more appealing over the past few years to the real-time and embedded software markets, which traditionally are far more conservative than the general application-integration market.

The technology-adoption life cycle curve also has implications for the newsgroup thread that questioned what SOAP's main technical contribution was. Participants in the thread denounced what they saw as a poor reinvention of a wheel that Corba had already perfected, but they failed to take into account SOAP's and Web services' appeal to the early market and early majority adopters, compared to Corba's appeal to the laggards and skeptics. Because SOAP and Web services borrowed ideas from other previous technologies including Corba, it's not surprising that there would be technical overlap. I view this not as a negative reinvention of the wheel but as the natural and positive carrying forward of good ideas and reapplying them to newer markets.

Consumer, Know Thyself

What type of consumer are you? Are you an early adopter, willing to try new things or take big risks and hoping to garner matching big rewards? Are you a mainstream majority adopter, waiting for someone else to work out all the bugs? Or are you a laggard or a skeptic, adopting new technologies only when it becomes obvious that not doing so will cost more than just going with the flow?

If you help make integration-technology purchasing or acquisition decisions for your company or group, it's important to know what kind of consumer you are, so you can properly assess where different technologies reside in their adoption life cycles and decide if they're right for you. However, it's just as important to realize that others in your company (or even in other companies that you might acquire or cooperate with) will almost certainly fall into consumer categories other than yours, and that whatever technologies you purchase or adopt might someday have to work together with technologies they choose. This phenomenon is one of the major forces behind the need for integration technologies, and it's unavoidable.

The end of middleware? I think not. ☐

References

1. J. Schwartz, "The End of Middleware," *Java Developer's J.*, vol. 9, no. 2, 2004, pp. 16–18.
2. G.A. Moore, *Crossing the Chasm*, Harper-Collins, 1999.
3. G.A. Moore, *Inside the Tornado*, Harper-Collins, 1995.

Steve Vinoski is chief engineer of product innovation for IONA Technologies. He's been involved in middleware for 16 years. Vinoski is the coauthor of *Advanced Corba Programming with C++* (Addison Wesley Longman, 1999), and he has helped develop middleware standards for the Object Management Group (OMG) and World Wide Web Consortium (W3C). Contact him at vinoski@ieec.org.