Cyon Research™

"The Innovator's Dilemma" in the context of CAx and PLM vendors – The case for an engineering software components market

A Cyon Research White Paper
January 16, 2006

Cyon Research Corporation
8220 Stone Trail Drive
Bethesda, MD 20817-4556 USA

phone: 301-365-9085
fax: 301-365-4586
Web: www.cyonresearch.com

"The Innovator's Dilemma" in the context of CAx and PLM vendors – The case for an engineering software components market

A Cyon Research White Paper

Executive Summary

Periodically, business books appear that have a transforming effect on the high-tech community. The first of these was "In Search of Excellence," the 1982 Peters and Waterman blockbuster that created the genre of business books. Geoffrey Moore's "Crossing the Chasm" (1991) was another.

In 1997, Harvard Business School professor Clayton Christensen released "The Innovator's Dilemma: When new technologies cause great firms to fail." It was fascinating—and for many high-tech marketers, depressing. The book highlighted issues that had never been so clearly elucidated, having to do with balancing innovation with listening to customers. It gave a strong jolt to the high-tech world—but only general hints as to how to avoid the identified dangers.

In 2003, Christensen and associate Michael Raynor dropped the other shoe: "The Innovator's Solution: Creating and sustaining successful growth." This is a how-to book par excellence, backed by extensive research, numerous case studies, and a down-to-earth presentation.

Cyon Research Corporation has explored the applicability of some of the ideas in "The Innovator's Solution" to the PLM industry. This paper is a summary of our findings, based on numerous interviews and discussions with industry leaders from both user and vendor firms, as well as with Christensen's research team.

(Note: While we summarize the relevant points of Christensen's observations, you will gain much more from this paper if you read The Innovator's Solution.)

To what extent can Clayton Christensen's theory of disruptive innovation, detailed in his book, "The Innovator's Solution," be fruitfully applied by CAx and PLM vendors? We have found it generally relevant and useful. For one thing, applying it to past disruptions and examining current conditions shows that it is reasonably predictive for CAx and PLM. That gives us confidence in applying it to current conditions in the hope of accurate forecasting.

For another, it is reasonable: When CAx and PLM were constantly pushing up against hardware performance constraints, monolithic software architecture was necessary for extracting maximum performance from computer hardware. Now that hardware designs have provided some "breathing space" in this regard, the benefits of modular software architectures come to the fore.

At the same time, the monolithic systems have added feature upon feature in successive releases, generally in the interest of product differentiation, such that they now greatly exceed what most users need or want. Many such unwanted features are perceived by users as “bloat,” that consumes the gains brought about by hardware improvements.

It therefore seems to us that modular architectures will soon dominate CAx and PLM software, and a substantial market for software components (and products based on those components) is about to come into existence. Software producers will find it more economical to acquire components and use their application knowledge to add value by constructing systems designed for the niches they serve. Just as operating systems and graphics drivers are no longer produced by application vendors, they will be able to focus their efforts on providing end-users precisely what they need, rather than devoting engineering to building that which will be available in component form.

"The Innovator's Dilemma" in the context of CAx and PLM vendors – The case for a robust market for components in engineering software

A Cyon Research White Paper

"The Innovator's Solution" says that there are two main mechanisms by which firms conquer and maintain control of their domains: Progressive improvement—"sustaining innovation"—and disruptive innovation. Each has certain conditions under which it is more successful than the other.

Definitions (adapted from Wikipedia.org):

> A **disruptive innovatio**n is a technological innovation, product, or service that eventually overturns the dominant technology in a market, despite the fact that the disruptive technology is both radically different from the leading technology and that it often initially performs worse than the leading technology, according to existing measures of performance. A disruptive technology comes to dominate an existing market by either filling a role in a new market that the older technology could not fill (as more expensive, lower-capacity, but smaller-sized hard disks did for newly developed notebook computers in the 1980s) or by successively moving up-market through performance improvements until finally displacing the market incumbents (as digital photography has come to replace film photography).
>
> By contrast, **sustaining innovation** refers to the successive incremental improvements to performance that market incumbents incorporate into their existing product.
>
> The theory of disruptive innovation: Christensen distinguishes between **low-end disruption**, which targets customers who do not need the full performance of the high end of the market, and **new-market disruption**, which targets customers who could previously not be served profitably by the incumbent.
>
> "Low-end disruption" occurs when the rate at which products improve exceeds the rate at which customers can learn and adopt the new performance. Therefore, at some point the performance of the product overshoots the needs of certain customer segments. At this point, a disruptive technology may enter the market and provide a product which has lower performance than the incumbent but which exceeds the requirements of certain segments, thereby gaining a foothold in the market.
>
> In low-end disruption, the disruptive company will naturally aim to improve its margin (from low commodity level) and therefore will innovate to capture the next level of customer requirements. The incumbent will not want to engage in a price war with a simpler product with lower production costs and will move up-market and focus on its more attractive customers. After a number of iterations, the incumbent has been squeezed into successively smaller markets; and when finally the disruptive

> technology meets the demands of its last segment, the incumbent technology disappears.
>
> "New-market disruption" occurs when a product that is inferior by most measures of performance fits a new or emerging market segment. In the disk drive industry, for example, new generations of smaller-sized disk drives were both more expensive and had less capacity than existing, larger-sized drives. Since size was not an important factor for the early computer market, these new drives seemed worse in every way. With the development of the minicomputer (or afterwards, the desktop computer, the notebook, and the personal music player), size became an important dimension, and these new drives quickly dominated the market.

Moreover, industries go through cycles with regard to products—cycles whose patterns are discernible, to those who ask the right questions. It's easy to understand why; once a critical set of challenges has been overcome, the next set in terms of importance becomes critical. In CAx/PLM, the initial challenge was automating previously manual activities. Once that challenge was addressed, the issue became the linking of the automation applications. Once that began to be taken care of, users began to demand better user interfaces, smoother connections among applications.

During this process of unfolding, the capabilities of each application area—graphics; analysis; CAM; and so on—were continuing to improve. For a time, users found that having a main CAD application and being able to choose "best-of-breed" add-ons for finite-element analysis, CAM, and the like, gave them the best overall results. But eventually, as the individual applications began to approach commodity status, the convenience and reliability of being able to buy everything from a single vendor came to the fore—echoing the old demand of "one system that does it all." Only now, vendors are finding that to support this demand, they must modularize their architecture—both to simplify maintenance and delivery of customer-tuned systems and to reduce their own development costs.

This research attempts to make use of the theory of disruptive innovation to explore the value proposition for component technology in the CAx/PLM marketplace.

Clarification of Terms

It is important to note that "tightly integrated" and "componentized" are not opposites. "Tightly integrated" refers to how software functionality is presented to the user—seamlessly, so that the user does not perceive that different functions are being performed by different systems or sub-systems.

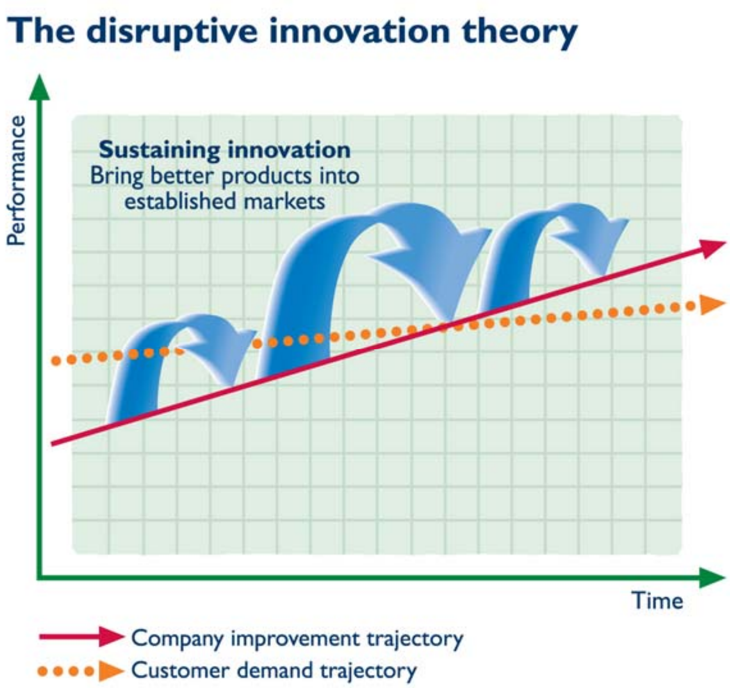
"Tightly integrated" contrasts with "loosely integrated," a term that characterizes systems whose user interface allows the user to see that invoking a function in fact starts up a separate program.

"Modular" and "componentized" are descriptions of software architecture, contrasting with "interdependent." "Modular" refers to software made up of modules, in which a module is in some way separable from the overall system, for independent testing and debugging.

"Components" are modules that have been further standardized in some sense, beyond the boundaries of an individual system, so that they can be used in other software projects. Conceivably, software components can be designed so that they can be used with components from other vendors in a single system, while still giving the system user the appearance of a tightly integrated system.

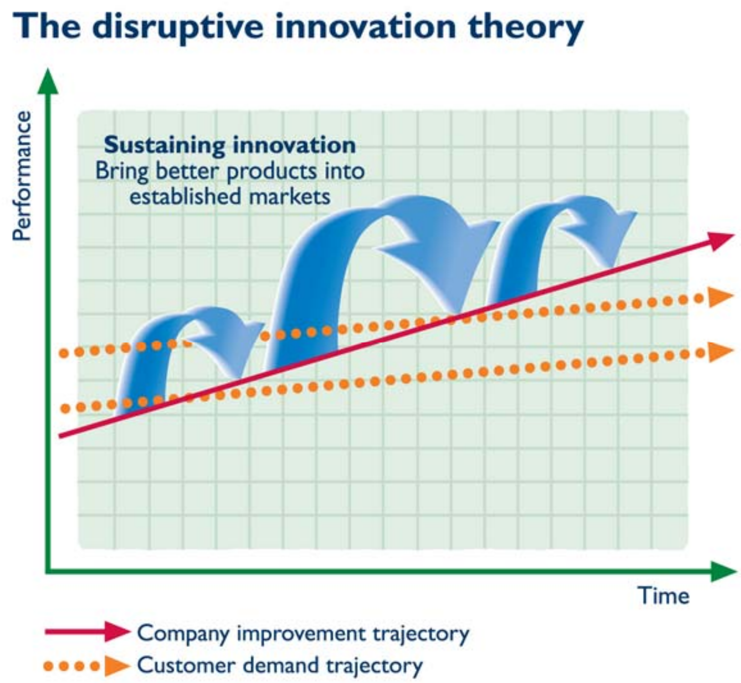
Partial history of disruptive innovation in the CAD market

The first diagram (below) shows, as a dashed line, an arbitrary set of performance criteria representing a need for speed, functionality, ease of use, etc., as perceived by the customer. The solid lines represent the customer's perception of what a vendor is delivering. Obviously, different customers will have different views of needs and performance. For an entire market (over many users), each of the lines represents the mid-point of a bell-curve.



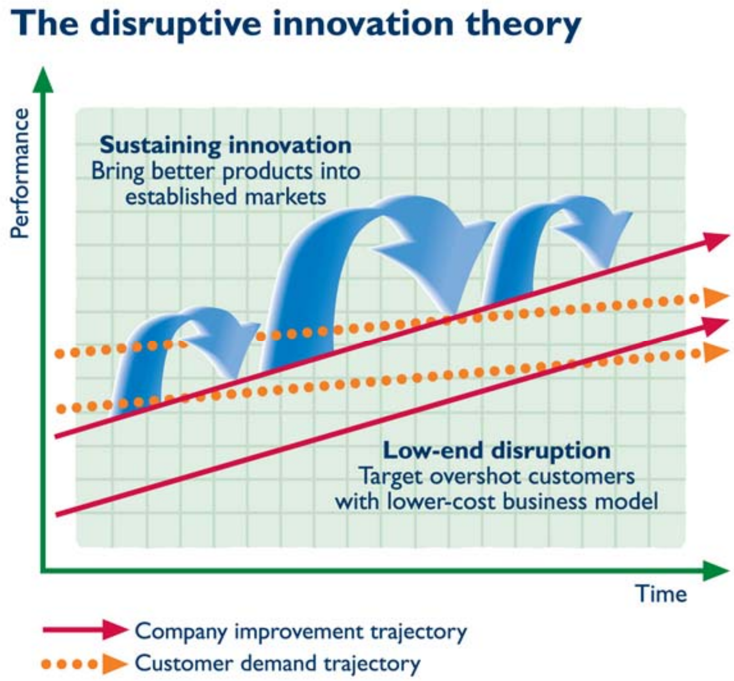
Consider this diagram as representative of the state of high-end CAD from the '80's on the left, where CAD systems did most of the job requested but still left some users wanting more, to where we are today on the far right, with CAD systems providing much more "performance" than even most high-end users can consume.

This next figure (below) shows a new group of users for whom the high-end systems were too expensive and for whom high-end systems were "overkill" – too expensive, too much "performance". (Note that while these users have an ever-increasing demand for "performance," the "performance" provided by high-end CAD is increasing at an even greater rate.



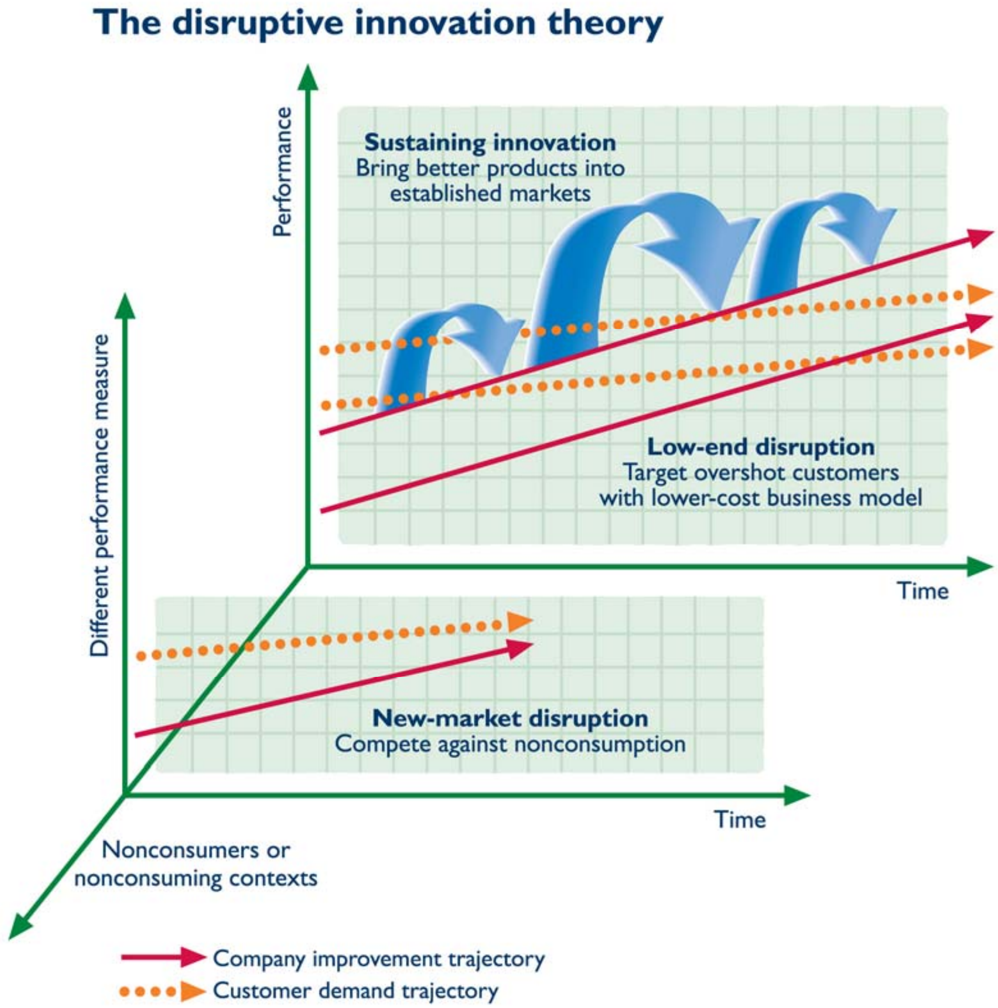
This performance demand of the new group of users began to be met by mid-range systems in the '90s (lower solid line in the figure below). These low-end disruptors, led by SolidWorks and Solid Edge (and eventually followed by Inventor), were able to reach a much broader audience than the high-end incumbents for two primary reasons: 1) They were more affordable, and 2) they were easier to use.

When these systems entered the market in the 90's, they didn't come close to meeting the "performance" requirements of high-end users. But today, most of those users' "performance" requirements have been met by the rising "performance" curve of that group of mid-range products. The effect of this has been—and will continue to be—an erosion of the base of users willing to pay a premium for high-end products. It has also resulted in a significant reduction of the average sale price of high-end products.



As mid-range products have continued to evolve, they have begun to exceed the "performance" requirements of their user base. At the same time, market leaders have recognized that there is a large class of under-served users who would benefit from some sort of interaction with the models created with both mid-range and high-end systems. To some extent, the entire PLM explosion can be viewed as an attempt to reach this broad market.

The figure below shows this as a new market disruption. Light-weight drawing formats are a key battleground of this new market, as are cPDM, drawing viewers, and several other classes of offerings that are going after this broad, but not yet differentiated, market.



Low-end Disruptions

In the CAx marketplace, full-function CAx products were disrupted at the low end—first by AutoCAD during the mid-1980s, and later, by mid-range modelers like SolidWorks and Solid Edge. In both cases, the market favored the introduction of new products at the low end, targeting non-consumers and over-served customers. Interestingly, it is possible that we are beginning to see this happening again—but now in the mid-range—with mid-range modelers becoming too complex and too full-featured for some—as yet untapped—segment of the market, paving the way for a new player to step in at the low end and present an offering that would appeal to non-consumers and the over-served customers in this space.

In order to understand how CAx/PLM software components are being used and might be used in engineering software, we spoke with consumers and producers of PLM component technology, both internal and external to the company.

There was also an attempt to substantiate findings, where applicable, in accordance with what Christensen refers to as the Law of Attractive Profits. As put forth by Christensen, the Law of Attractive Profits suggests that when modularity and commoditization cause attractive profits to disappear at one stage in the value chain, there is usually an opportunity to earn attractive profits with products that have not become commodities at an adjacent stage. As an illustration, publisher and industry observer Tim O'Reilly notes:

> 'IBM introduced this open architecture for a standardized personal computer in 1981, and we built this whole industry that was based on people assembling all the commodity components. In the old days, building a computer was hard. You designed your own, you wrote your own operating system and once they released those specs they thought, "Wow!"
>
> "It was just a short-term tactical move [for IBM], but it changed the entire industry. It took the power away from the hardware vendors and it pushed value up in ways that they didn't understand so that Microsoft ended up with a lock on a very valuable layer up the stack. That's the law of conservation of attractive profits. Building computer hardware [now had] a lower margin…and value went up the stack.
>
> "But there was also a discovery that Intel most notably made…: if you could figure out how to be the single source for a really valuable component in the system, you could also get enormous power. So, value went down the stack as well. So, the middle layer, the assembly of this commodity hardware, became relatively low-margin and the businesses below and above became very high-margin. That's Christensen's law of attractive profits [in action]." (*Transcript of interview with Doug Kaye, "ITConversations," at O'Reilly Open Source Convention, 7/26-30/2004.*)

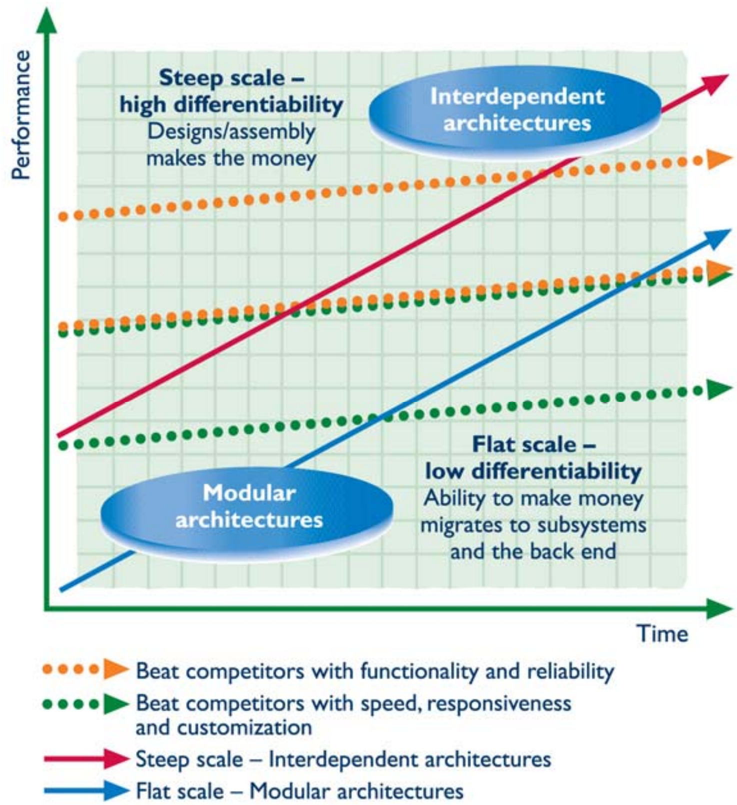


Figure 1. Attractive profits typically are earned in the stages of value-add in which non-standard integration of components occurs. (Source: Clayton Christensen, "The Innovator's Dilemma")

In the CAx/PLM market, we are beginning to see a shift toward customers demanding task-specific products and solutions. We see CAx and PLM following a similar path, moving from general-purpose tools and offerings to ones that are more specialized. The innovation curve crosses an inflection point, and then monolithic products become "good enough;" at that point, the advantage shifts to modular products. As the modular products continue to add more generic functionality, they start to over-serve the market. So they strip out generic functionality in order to serve narrower niche markets.

Again, the advantage here goes to those employing a modular architecture, because they can respond more quickly to task-oriented customer demands than those with highly integrated architectures. This puts component developers in a favorable position.

But it is also important to point out that where modularity can provide an advantage at one point in the value chain, interdependent solutions are advantageous in another.

Is Commoditization Happening?

In discussing how Christensen's theory of disruptive innovation applied to the CAx/PLM sector, we asked Christensen's team, "We are seeing some commoditization in the CAD marketplace—customers are, for example, complaining that they don't need more features, that they can't handle what they've already got; and that CAD products were all beginning to look alike. Such conditions suggest that customers are being 'over-served,' in terms of Christensen's model. However, is it possible that a customer's request for ease of use and fewer features could mean that the customer is *under*-served –and therefore, that we might begin to see the advantage shift to providers offering a more tightly integrated solution again?"

Interestingly, the team representative said it could. He indicated that markets can and often do fluctuate in this regard—benefiting initially from interdependent architectures, then later from more modular ones, then during what might be viewed as a realignment phase—they can once more benefit from interdependence, and the cycle begins again.

Translated, this could mean that in the CAx/PLM marketplace, we may be entering a phase in which customers stand to benefit from more tightly integrated solutions again. But in contrast to earlier monolithic solutions, those that seem to be succeeding now take the form of component-based applications that are targeted to customers' specific needs. This is evidenced by an increased interest in "best practices" and industry-specific PLM solutions targeting aerospace and automotive, as well task-specific CAx tools designed to address the design requirements of, say, mold-makers, airplane-wing designers, or engine designers.

Another way to look at this is to think of CAx as entering a phase in which targeted solutions are likely to prevail, making it an attractive market for those software providers that can deliver such solutions quickly and easily. Such providers will benefit from having offerings that consist of components that can be easily packaged and re-packaged to provide a solution to match the customer's needs—that is, to match the job to be done.

Monolithic or Modular?

Today's component-software technologies permit the creation of systems that appear seamless to the user, yet are in fact made from components. The demand for tightly integrated solutions simply reflects a growing sophistication in the market; users have become less and less willing to tolerate the performance reductions and user-interface irregularities of poorly integrated components.

With respect to the emerging PLM market, we see modularity and componentization benefiting software development as well, but for different reasons. Here, the market is immature; thus, customers have benefited from full-function, tightly integrated solutions—an environment that typically favors overall operational performance over that of individual subsystems or modules. However, even in this situation, componentization enables software providers to quickly deliver a solution that addresses the requirements of its customers by facilitating the interoperability of the individual segments that comprise the PLM solution. And today's component technologies puts performance penalties for this approach below the threshold demanded by the market, in most applications; that is, the loss of performance caused by employing component architecture is not noticeable to the user.

Moreover, as enterprises take more and more advantage of the collaborative aspects of PLM, component-based systems gain in favor; they are easier to deploy and maintain in the distributed complexity of inter-company operations.

Since there is a large segment of the market that does not require full-function, tightly integrated, high-performance PLM, there is an opportunity for a solution provider to step in at the low end of this market, providing only the most basic capability in terms of managing product data and facilitating collaboration—and in so doing, potentially succeed in disrupting the PLM marketplace.

Bottom line: CAx and PLM can both benefit from componentization of software solutions—CAx because it is mature, and customers want highly targeted solutions; PLM because it is immature, and customers value overall system integration and performance. Component software provides a more effective basis for addressing both of these markets than the more-traditional, less-modular approaches to system architecture—especially given the current demand for more industry-specific solutions.

In a given hardware context, interdependent software architectures typically offer higher performance than do modular architectures. The reason is simple: There is a certain amount of computational overhead associated with modularity that does not exist in "all-in-one" architectures.

However, that statement is simplistic. So much depends on the quality of programming in both architectures that it would be misleading to say that an application with an interdependent architecture will always outperform a similar application built on a modular architecture.

Also, Moore's Law is a tide that raises all boats. In many areas, modular architectures whose overhead was prohibitive a few years ago have won out over interdependent architectures because computational power is now abundant and memory is cheap.

Just as it is much easier to turn a fleet of small vessels than it is to turn an aircraft carrier, modular architectures enable software vendors to respond rapidly to changes in the market. The value of modularity in a commodity market is in the "mix and match" power it gives to software vendors, to assemble task-targeted systems out of modules and thereby come up with products that address niches quickly and cheaply.

While the general trend in software is toward modular architectures, especially for large systems, vendors with software products that are not so constructed would be justified in questioning whether their product architecture ought to be radically redesigned for the sake of keeping up with trends, or even for improving maintainability.

This is precisely why insight at the level of the Christensen book is so helpful. It enables us to generalize lessons learned in other markets and apply them to ours.

Once a product reaches "commodity" status in its market, the principles uncovered and enunciated by Christensen and his team indicate that companies whose products are modular take the next round. They are able to serve end-users and vendors who can rapidly configure targeted products for niches with clear but narrowly defined needs

Software architecture, like all architecture, is a mixture of engineering, science, and art. It is a balancing act: To deal with the challenges of system complexity, self-contained functional modules that can be designed, debugged, and maintained on their own, are indicated.

But to wring the greatest possible performance out of hardware, a system architect may choose to eliminate the overhead of inter-module communication by creating a monolithic system.

If we add to our list of desiderata the requirement that modules be re-usable in other systems, we arrive at the reasoning behind software components—independent software modules that can be assembled to create systems. Thus, to clarify our terminology: Components are modules, but not all modules qualify as components. Components are modules designed to be reusable outside the narrow constraints of a given project.

Software system vendors are drawn towards components, to minimize the costs and risks of development and maintenance. End users demand maximization of performance, up to a certain level of functionality. Once hardware power makes the performance differences between monolithic systems and component-based systems invisible to the end user, the maintainability and extensibility of component-based software lead to their dominance—and to economic opportunity to the suppliers of components.

Winning Software Strategies

So one winning strategy for CAx/PLM vendors is to focus on componentizing their products for rapid response and deployment. This entails modularizing them and packaging them in a generally reusable form, so that they can be easily employed with other components to create new systems or enhance existing ones.

Another winning strategy for vendors is to develop and market components. Since the development of components will be beneficial for their own products, as well, it does not constitute a high-risk shift in business direction. And it opens the possibility of acquiring components from other companies to round out their systems, rather than having to develop them themselves.

Christensen also makes the observation that emerging markets, like PLM, value "all-in-one" systems that are highly integrated and have good performance. However, while PLM per se is an immature market, it has been a long time in formation. Joseph Harrington's "Computer-Integrated Manufacturing" described much of what PLM now does—in 1973.

As a result of this gradual unfolding of automation within the manufacturing organization, individual areas differ greatly in the degree to which they have been computerized. Many are quite sophisticated, although most lack system-level integration.

Furthermore, the increase in outsourcing and "coopetition" necessitates a greater level of communion among automation systems, both within and among enterprises.

For these reasons, it makes sense to consider the next phase of PLM as one of successful integration of components, rather than deployment of monolithic highly integrated systems. It is the interfaces and the overarching control systems that are flawed or lacking. These needs can best be met with component architectures.

Summary and Conclusion

Our question was, "To what extent can Clayton Christensen's theory of disruptive innovation be fruitfully applied by CAx and PLM vendors?" Our research indicates:

- The theory of disruptive innovation makes sense in the CAx and PLM markets, in that applying it to past disruptions yields results that match present market conditions;
- In the current computational environment, the performance advantage of monolithic over modular software architecture is no longer significant, for most CAx and PLM applications. Thus, the economic and reliability advantages of modular architecture should now be dominant considerations in engineering software system design;
- Market conditions are therefore favorable for engineering software companies to capitalize on their investment in software through the creation and sale of software components. The indications are that modularity is not a temporary trend in software architecture, but a major and permanent one. It makes it possible for software vendors to profitably customize their products for market niches.

About Cyon Research…

Cyon Research is a consulting firm that provides design, engineering, construction, and manufacturing firms with a strategic outlook on the software tools and processes they rely on to create the world around us. Cyon Research also supports the vendor community with its unbiased insight, vision, and expertise to help them understand the complex nature of their markets and grow, by serving the needs of their customer base.

Cyon Research brings to its clients a unique combination of experience, perspective, and insight, supported by an extensive network of well-established industry relationships. Our close contacts throughout the user, analyst, vendor, and developer communities provide surprising benefits for our clients and add significant value to our services.

Those relationships are enhanced by our publications and events. While consulting is the heart of our activities, our publications and websites—including *CADCAMNet*, *Engineering Automation Report*, and *CADwire.net*—are our voice. Through them, we connect daily and monthly with the user and vendor communities. And COFES: The Congress on the Future of Engineering Software, our annual, invitation-only event, is our face—the place where we can make the types of connections that just aren't possible through any other means than face-to-face.

The focus of our research within the realm of design, engineering, construction, and manufacturing is technologies and markets that are likely to become real within the next two to six years.

The domain of our research is the tools, processes, and procedures used in the design, engineering, management, and production of the built environment and manufactured goods.

Funding for this white paper was provided in part by UGS. Watch CADwire.net for additional Cyon Research white papers.

Cyon Research™

Cyon Research Corporation
8220 Stone Trail Drive
Bethesda, MD 20817-4556 USA

phone: 301-365-9085
fax: 301-365-4586
Web: www.cyonresearch.com

