

Integrated eClinical Solutions Be Afraid, Be Very Afraid

By Timothy Pratt, PhD

Why are lots of EDC vendors talking about how integrated their systems are? And what's this new buzzword emerging - eClinical? Is this just EDC? Why should you be concerned about an "integrated system" - surely that's a good thing, right? Let's start with a few definitions and move on to discussing the important differences between integrated and homogenous technologies, including the impact to an organization and clinical success.

eClinical has emerged recently as a term used to encompass the shift in clinical studies to electronic computer-assisted capture and management methodologies to increase efficiencies and quality by replacing manual and paper-based systems. Electronic data capture (EDC) is just one component of the eClinical universe, which also includes clinical data management systems (CDMS), clinical trial management systems (CTMS), interactive voice response systems (IVRS), patient reported outcomes (PRO), and so on.

Integrated (Heterogeneous) Systems

Integrated eClinical offerings are comprised of a set of software applications that were usually developed as proprietary standalone applications, and have subsequently been forced to "talk" to each other - that is, to make available what they "know," in whole or in part to the other elements of the system. In computer parlance, these are known as heterogeneous systems - the various software "islands of automation" that make such systems up were never intended or designed at inception to have common knowledge of the data contained within each other island; they were built independently. The islands thus have to have bridges built between them to communicate, which limits data traffic to the design and location of the bridge, not to mention the gridlock encountered if the bridge falls down.

Homogenous Systems

Homogenous eClinical systems differ fundamentally from integrated systems in that they are one entire "thing" and fundamentally consist of entities that share common knowledge of one another's data structures, formats, and source code. There is one underlying software framework and code base that is the engine that drives all the offerings of the system, be it EDC, CDMS, CTMS, IVRS, PRO, and/or any combination thereof. Because homogenous systems don't have to "discover" the data formats during processing

they are significantly more efficient and stable than integrated - heterogeneous systems; this also often reduces the need for future additional supporting software purchases.

Well-designed homogenous systems also tend to have far fewer issues interfacing with other data sources, such as diagnostic devices or financial software, if they are based in open-source code, which, by its nature, can be easily modified and effect interoperability via common transfer protocols

Examining Integration

By now, you may well be asking yourself the question, "So what's the difference in practice?" The difference is probably best answered by examining what the intent of integration is, and why it is often less than successful.

The whole purpose of integrating systems is to increase efficiencies by making related software applications' data visible and usable to the system as a whole. The problem is that this is extraordinarily hard to do. Ask an IT person, who will tell you it is often very difficult to get disparate systems developed in isolation to reliably "talk" to one another. What this means in practice is that the integrated solution is often problematic, doesn't work as advertised, is susceptible to crashes and bugs when elements of the individual packages are updated, and so on. In addition, the promise of real-time data visibility and operations management often is not delivered in integrated solutions. Because the package components run as separate applications under a broad umbrella, the performance of the system as a whole is necessarily dependent on the lowest common denominator's performance. Many of the packaged applications were never designed to be interoperable with other systems; they run at different speeds, process and output data at different rates and in different ways, and generally behave independently rather than as a cohesive whole. Such component behavior

Originally published in
PharmaVOICE VIEW on E-Solutions, October 2006



mednetstudy.com

is intrinsically inefficient and often requires periodic outside (human) intervention to control, obviating the automaticity and subsequent efficiency desired in the first instance.

The Whole is Greater Than the Parts

A homogenous system, on the other hand, is a single software application that supports many functions. It doesn't need to be "integrated" because it is, by definition, one whole thing from the start. Thus, a homogenous system has no issues with regard to making data available for different functions required, and updating one element updates the whole system. As a consequence, a homogenous system tends to be much more stable and less problematic. Other benefits include load balancing, inherent redundancy, simpler system design, and lower system maintenance costs. All of which is geekspeak for "it's simply a better way of doing things."

So why are so many vendors talking about "integrated" systems? One reason could be that, lacking a flexible underlying homogenous framework, they have no other method by which to offer a full eClinical solution. Apart from day-to-day operational performance, customers have a problem if they purchase an eClinical system from a vendor that does not possess one or more elements they're looking for (say, honoraria payments); in which case, they're out of luck until that vendor buys a company with such a tool. Then the customer has to go through the lengthy process of trying to integrate the system – after the often painful process of integrating the acquired company into its culture; no mean feat in itself.

Recently, a representative of a major vendor, which has grown by acquiring independent software packages and "integrating" them, summed up his company's problem by stating that EDC will not be able to handle budgeting, investigator payments, schedule monitoring, etc.¹ And he's right. The company would need to acquire and attempt to integrate another program to handle those tasks.

Currently, there are advanced homogenous eClinical solutions that can handle honoraria, scheduling, financial package interoperability, inventory control, randomization, IVRS, PRO, COMS, CTMS functions, and much more. They also happen to do EOC and do it very well. The impact to your organization can be extremely positive with such systems.

What to Look For in an eClinical System

If you are in the market for an eClinical solution, and have never bought a product of this type before, there are some indicators to watch for. If the system touts that it is "integrated," that's an obvious giveaway as to what is being

offered. Other cautionary trigger words are "suite" and/or specific brand names for each element, such as EDC or CDMS. If a system looks attractive, ask the vendor some pointed questions about how the individual elements were designed and communicate with each other. You might need to ask the technical folks rather than the sales personnel. It's very important to speak to users in other companies and not just the ones the vendor directs you to. After all, they're hardly likely to recommend you speak to someone with a negative opinion, right? You might find the technical folks – IT, IS, and so on – at the other company a little more forthcoming on the ease of operation and maintenance.

eClinical can be a wonderful thing. The promise of more efficient processes, of cleaner data, faster, and generally accelerated clinical timelines are very alluring. Just make sure when you're examining what's out there you don't get shown a picture of a supermodel and find yourself lumbered with Frankenstein's integrated monster: a whole lot of bits and pieces sewn together from various sources. That story does not have a happy ending

¹ *Fisbach D, PhaseForward, "Does EDC herald the end of CTMS?", Bio-ITWorld. 20 July 2005.*