Overcoming the real challenges of integration

With technologies such as EDC, ePRO and CTMS becoming commonplace in the clinical trials arena, the next generation of electronic systems needs to tackle the problems of data sharing. **David Stein** reports

KEYWORDS: Integration systems; Audit trail; Electronic data capture (EDC); Good programming practice (GPP)

The adoption of clinical technologies has been the key to running clinical trials more rapidly and efficiently. As more systems are brought into the clinical trials arena, there is often overlap in the data they use. For example, patient demographic data may reside in databases in electronic data capture, interactive voice response and clinical trial management systems, among others. To eliminate redundant data entry and work toward achieving interoperability of these systems, many biopharmaceutical companies have taken advantage of integration.

The key benefits of integration are well documented.[1–3] In brief, automated sharing of data eliminates the need to enter the same information into multiple systems. This obviously saves time, but more importantly, it eliminates possible discrepancies between databases. Whenever discrepancies are possible, extra quality assurance steps are necessary to resolve any mismatches. And, because data cleaning and discrepancy resolution are often performed late in the course of a study, these activities may delay locking the database for analysis. For example, the relatively simple task of reconciling demographic data between databases for electronic data capture (EDC), interactive voice response (IVR) system, electronic patient reported outcomes (ePRO) and lab results invariably identifies inconsistencies that must be rectified. Even if it is not necessary to query the investigative site, such reconciliation takes precious time and resources.

Integration issues

Despite the benefits of connecting systems, new challenges have appeared. Each integration project is essentially another technical application requiring testing, validation, maintenance and support, including help-desk support. Furthermore, many of these connections have been specified and programmed by technical members of clinical teams rather than developers or engineers. Although such technical people are well positioned to create the user requirements and perhaps even the specifications, there are other important considerations. While engineering departments typically have robust standard operating procedures (SOPs) governing the software development life cycle (SDLC), this is not usually the case for groups building *ad hoc* integrations.

So, what guides integration development? Are all integrations built according to industry standards for computer systems validation? Are proper change control guidelines in effect? And if more than one vendor company's software is being connected, do all vendors support the integration? While many integration applications satisfy the foregoing requirements, some connections have fallen short. Help-desk staff are expected to be properly equipped to quickly resolve all incoming issues, but, unfortunately, the help desk is one of the most commonly overlooked areas when integrations are built outside of a proper SDLC environment. The result is slow issue resolution and frustrated users.

Another problem with programming connections outside of the engineering department concerns change control and software upgrade paths. For example, when a vendor recently upgraded their software for a sponsor they performed a robust series of tests, along with the sponsor, to ensure their upgrade would function as expected. The new software was then loaded into production on a live study according to standard change control procedures. However, both the vendor and sponsor neglected to test all the integration applications touching the software and at least one connection was broken. From an auditor's perspective, this situation is unacceptable. Clearly, any applications that may be affected by system or software changes, including integrations, must be part of any upgrade or change control plans.

This case example also brings other compliance issues to light. Consider the ripple effects caused by the broken link between systems. When the error was eventually caught and the systems were reconnected, there was a backlog of data that had not been transferred. In some cases, where batch or asynchronous connectors are used, the backlog may automatically be resolved as soon as the systems are reconnected. In this case, though, the connection was built to support real-time data exchanges, not batch transfers. This meant that the vendors had to create a new procedure to move the backlog of data between their systems. While in such a situation there is the temptation to do a 'quick fix,' analogous to

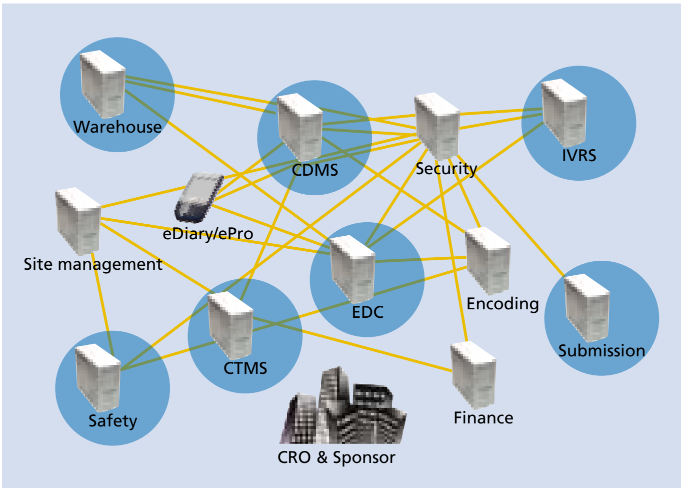

Figure 1: Point-to-point integration solutions require multiple linking applications that must be maintained and supported in a compliant environment.

performing a series of manual cut-and-paste operations just to get the job done, doing so would raise compliance issues and more importantly, could cause additional errors. It is also clear that any solution to resolve this backlog would require a well-documented plan to comply with GCP; thus, a 'quick fix' would violate GCP. Such manual intervention would also bypass the audit trail.
The audit trail is a vital part of maintaining a validated environment and is one of the key compliance issues to be reviewed when creating integration links. Even if the independent systems maintain their own audit trail, it is important that integrations too are audit trailed. Questions must be asked: Is all the information being imported placed in the audit trail along with the source of the data? And do data exports require such logging as well? (Perhaps this is not required in all situations, but was this question even considered by the teams doing the integration?)

The issues just described above are not unique to integration applications, but they certainly are much more prevalent in these projects, especially when they are built outside of a real development environment according to good programming practice (GPP). Also, as connections proliferate, so does the number of point-to-point applications (see Figure 1). Any given system may be linked with four or five other systems, thereby multiplying the likelihood that one or more of the issues described will occur.

New models of integration

To address the validation and compliance challenges discussed, as well as many technical issues, a new model for integration is emerging. This follows the middleware concept used in several other industries, but is geared towards the clinical environment, with features such as lab import tools, Clinical Data Interchange Standards Consortium (CDISC) export and the like. Clinical middleware is a platform or hub that acts as the centre of the universe for all systems that need to communicate among themselves (see Figure 2).

This clinical hub is a true software product built according to GPP and with a well-documented SDLC. Each application 'talks to' the hub via a connector, which is an application built by engineers or developers. Connectors, once written and tested in a validated environment, can then be configured for specific studies by non-programmer personnel using a simple interface similar to a wizard. This clinical middleware method, like any configurable application, provides a more repeatable, consistent process and requires far less validation effort than one-off integration programmes. It is also far more reliable.

It has other advantages, too, over traditional point-to-point connections, as the clinical hub manages all interactions between systems. More specifically, it catalogues all systems it touches and not only 'knows' what data each system contains, but keeps track of what information each system needs from the others.

A good example would be to review what might happen when a patient is randomised in an IVR system. As soon as randomisation takes place, the clinical hub is aware of the event. It also sees that a connected EDC system needs the randomisation and kit number assigned and it accesses these fields. The hub then maps the fields into the format and names the EDC database required, as identified in the EDC connector, and sends the data immediately. The data can now be viewed on an electronic case report form (eCRF) in the EDC system. Next the hub sees that a connected clinical trial management system (CTMS) needs to be made aware that another patient has been randomised and follows a similar process to inform the CTMS. In like manner, any other connected systems can then be given the relevant data triggered by the randomisation event. Yet the IVR system does not require multiple connections to communicate with each of these applications; rather, it has only one connection – to the hub itself. In the event of a software change or upgrade for any given system, only one connector must be reviewed and/or changed to maintain all the interactions between applications.

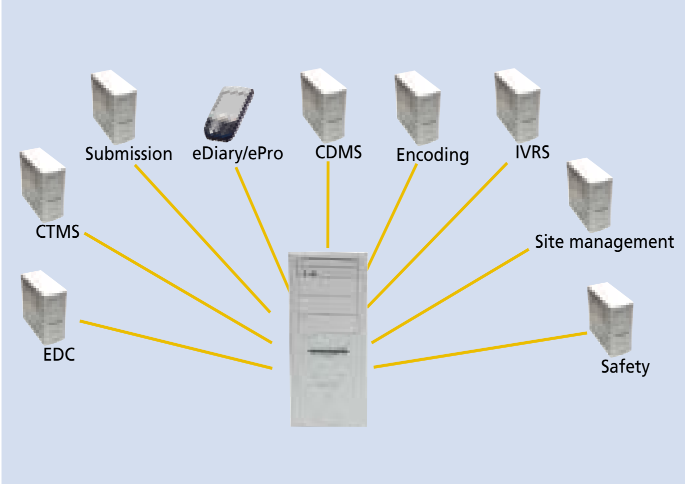

Figure 2: The clinical middleware solution, sometimes referred to as clinical technology integration platform (CTIP).

Meanwhile, in the above randomisation scenario, the clinical hub maintains a full audit trail for every integration and, in fact, every piece of data moving between any two systems. Furthermore, the audit trail format is consistent and centralised, unlike most *ad hoc* applications.

So, why not just purchase an off-the-shelf middleware product? Firstly, most generic middleware products have a high price tag. Secondly, a generic middleware hub will not come with any connectors ready to plug into the clinical trial environment. Such systems will not have any inherent capabilities to help users comply with industry specific requirements, such as those contained in FDA 21 CFR Part 11 or FDA guidance on Computerized Systems Used in Clinical Investigations (CSUCI), or standards like CDISC. For these reasons, many companies in the vendor community, including ClinPhone, Datatrak, Medidata, Omnicomm and Phoenix Data Systems have been advancing their integration capabilities based on the clinical middleware concept.[4]

Conclusion

No doubt, at this moment, there are hundreds of study-specific applications sharing data and talking to one another. The majority of these interactions are taking place using point-to-point applications that may never be used again. However, an increasing number are using the middleware concept and are seeing the advantages of having a compliant, repeatable process, complete with a full audit trail that can stand up to the scrutiny of an experienced auditor. Already, clinical middleware applications and connectors are being built by engineers using a solid SDLC in a validated programming environment.[4] And the trend toward clinical middleware is growing in order to replicate the successes seen in other industries. A few years from now, it will be commonplace.

References

1. R Case, 'In Search of the Holy Grail – Chasing ultimate clinical efficiency, one small step at a time', *Pharmaceutical Executive,* July 2005.
2. B Harper. 'Meshing EDC with CTMS,' *Bio-IT World*, February 2007.
3. J Mcllwain. 'A to Z Trial Integration,' *Applied Clinical Trials*, October 2007.
4. L Ramos. 'The Promise of the Next-Gen eClinical Trial Software', Forrester Research Market Overview, 15 March 2006 (http://www.datatrak.net/company/publications/The%20Promise%20Of%20Next-Gen%20eClinical%20Trial%20Software.pdf).

