

Centralized or distributed? Some principles in implementing administrative systems, with examples from the KU Medical Center

Jim Bingham
Associate Vice Chancellor for Information

Resources

University of Kansas Medical Center

When considering the implementation of a key administrative system (for example, a financial system, a human resource system, a payroll system, a student administration system), 4 principles need to be kept in mind:

1. There is an *inverse* relationship between the span of a system (how many functional entities are served by it) and the granularity of the information it provides.
2. There is an *inverse* relationship between the span of a system and the degree to which it may be customized to meet the specific functional needs of each entity.
3. The acquisition of and implementation of a software package is frequently confused with the development of a functioning system of which the software is an integral part but only a part.
4. Thus, while implementing a system at a relatively high level (for example, statewide), may result in less cost than implementing multiple systems at a lower level (for example, individual university level), the actual functionality provided by that implementation is substantially less and, thus, its value to the entities that use it greatly reduced.

Here's how these principles play out at KU Medical Center.

At KU Medical Center, we've run the PeopleSoft Human Resources/Payroll system since 1995 and PeopleSoft Financial System since 1999. (We also share the PeopleSoft Student Administration System with the Lawrence campus, but that's a story for another day).

We have created more than 100 "bolt-on" modules for our PeopleSoft systems which provide various types of functionality not provided by the PeopleSoft software (note bullet 3 above). This functionality includes:

- Budget module
- Leave accrual
- Time and labor reporting
- Phased retirement reporting
- Compliance module
- United Way

- F & A (for federal Indirect Costs associated with federal grants)

And many, many others.

We have built interfaces into at 11 other systems which provide various kinds of special functionality and then connect to PeopleSoft for added functionality. These systems include:

- Facilities Management
- Laboratory animal management (Sirius—see below)
- Micro Core Facility
- Information Resources
- Biotechnology
- Telecommunications management (Kansas City campus)
- Telecommunications management (Wichita campus)
- MIP (a KUMC Research Institute system)
- Graduate Medical Education (Akcia—see below)
- PeopleSoft Student Financials (financial information from the PeopleSoft Student administration system housed at the Lawrence campus)
- PeopleSoft Payroll

Here are two examples of how these interfaces work:

1. We use the Sirius system to manage our extensive and highly regulated inventory of laboratory animals. Sirius provides functionality including protocol management, animal procurement, cage card generation, animal and cage census, diagnostic lab & pathology, animal records, personnel training tracking, and associated financials. We connect the Sirius system to PeopleSoft so the financial transactions it handles at the detail level are incorporated into the University's larger accounting and receivables transactions and, thus, managed as a sub-area of the University's budget.
2. We use the Akcia system to manage our Graduate Medical Education (that is, Residency) teaching and clinical care program. Akcia provides scheduling of Residents, duty hours tracking, clinical activity tracking, reimbursement (that is, from Hospitals who use our Residents), and many other features. We take the information about how and where the Residents spend their time; do some magic to it in our data warehouse that converts it into billing information; and then import that information into our PeopleSoft Financial system where we generate bills; track receivables; age the accounts, and (ultimately) recognize revenues and create high-level reports.

Not only have we incorporated more than a hundred locally-developed modules into our PeopleSoft systems and built connectors to other local systems, we have also implemented PeopleSoft modules which state agencies would derive no benefit from. We implemented the Grants Module to enable the KUMC Research Institute—a private, not-for profit entity—to manage grants within the PeopleSoft

framework. And we have integrated the Research Institute's grant accounting into our PeopleSoft Financial system so we can get the "big picture" of financial information.

Despite this kind of customization, some of our schools and departments have discrete transactional and information needs that PeopleSoft cannot accommodate. There are literally dozens of "shadow" financial systems in use that provide essential operational functionality because it is cheaper to build the shadow systems than to customize PeopleSoft to provide that functionality. Generally these systems either provide more detail than PeopleSoft or they provide specialized reporting required by auditing, regulatory, or accrediting agencies. These systems complement our PeopleSoft system in a way not very different from how our PeopleSoft systems currently complement the State's payroll system. They are not redundant and they are not wasteful. They are essential parts of our educational, research, clinical and administrative business operations.

So we have taken the PeopleSoft software and used it to build a financial system (bullet 3 above) that meets innumerable unique and essential business needs of the KU Medical Center. We have augmented it with other systems where appropriate to create an elaborate web of computing systems that perform essential business operations. We prefer to modify PeopleSoft when possible because we benefit from the integration, but the cost/benefit equation drives the outcome.

A financial system implemented at the state level may very well meet the state's high-level needs for financial management and reporting but it will be woefully ill-developed to meet the University's needs ("inverse relationship" bullets 1 and 2 above). There is no way a state-level system can provide even a fraction of the functionality provided by systems distributed at the university level.

And, even if it were able to do so, it would take decades for it to actually get done...decades better spent building next-generation administrative systems *for the State* instead of redeveloping and re-implementing less effective versions of systems already built at the University level.