Approaches to RFPs for Student Information Systems

Jeffrey von Munkwitz-Smith, Ph.D.
University Registrar
University of Connecticut
136 Wilbur Cross Building, U-77D
Storrs, CT 06269-4077
jvon@uconnvm.uconn.edu

Abstract:
Requests for Proposals (RFPs) from several institutions were analyzed. Common elements were identified and several models for describing functional elements were identified:
- highly detailed
- lightly detailed
- only overall objectives presented

These approaches are contrasted for the registration function and another possible model, scenario-based, is suggested.
Approaches to RFPs for Student Information Systems

Background

On my third day of work at the University of Connecticut, in January 1997, I was at a meeting to discuss replacement of the institution’s student information system. Not too many months before I had been involved in creating a Request For Proposal (RFP) and selecting a vendor for a new student information system at the University of Minnesota. To quote the famous Yogi Berra-ism, it was “deja vu all over again”.

Minnesota’s RFP was a long document--several hundred pages--with a functionality section that contained a great deal of detailed requirements. It was created by a committee of six people, with many more involved in contributing information and reviewing the document. It involved a great deal of effort, and a certain amount of frustration. The vendor responses averaged 10 pounds of paper. The review of the responses involved a great deal of effort and more than a little frustration. Some of us found it difficult to distinguish between vendors based on page after page of “Yes” responses to questions about functionality. (Digging a bit deeper, “Yes” was more than occasionally “Yes, but...”.)

Knowing the effort that was involved in creating Minnesota’s RFP, I got interested in what other models there might be for Student Information System RFPs. While working on the one at Minnesota, I had seen one from Washington State University. In contrast to the approach we were taking, Washington State’s was quite spare. It was only 31 pages long and contained only a brief, and quite general, description of the desired functionality. I sent messages to a couple discussion lists, CUMREC-L and REGIST-L, asking if people were willing to share copies of SIS RFPs and to comment on their effectiveness in eliciting understandable responses from vendors. People from several institutions responded.

There were documents, like Minnesota’s, which contained a high degree of detail. An earlier (1982) RFP from the University of Connecticut was two inches thick. Others, like Washington State’s, presented only overall objectives. Some might be characterized as “lightly detailed”, specifying only functional elements considered critical by the institution. An interesting one, from the University of Massachusetts - Amherst, included some short scenarios in its, highly detailed, functionality section.

Models

I analyzed RFPs from the University of Connecticut (1982), the University at Albany, the University at Binghamton, the University of British Columbia, James Madison University, the University of Maine, the University of Massachusetts - Amherst, the University of Minnesota, Oakland University, the Rhode Island System of Public Higher Education, Stetson University, and Washington State University. Some were for integrated administrative systems for the core student, financial, and human resource systems, but most of them were solely student information in scope. All but the University of Connecticut’s were issued in the 1995-1997 time period.
One of the issues that interested me was how the desired system functionality was represented in the various documents. A complete analysis of all eleven RFPs would be quite time consuming to undertake and too long to present here, so I focused on the sections on Registration, an area of particular interest to me as a registrar.

These sections varied considerably. The document from Washington State University listed 10 desired functional elements; it did not ask for a description of how the proposed system would accomplish those functions. The RFP from the University of British Columbia described the current registration process and included information on response times and transaction volumes, but did not include more detailed questions about functionality. Two types of questions were asked in the other documents: those asking for a “Yes/No” answer, sometimes with a space for explanation or an indicator whether the functionality was planned for a future release of the system, and those asking the vendor to “describe how” the system performs a specific function or handles a specific situation. For purposes of this analysis, listings of desired functionality were included in the “Yes/No” category.

An analysis of the questions asked about registration uncovers surprisingly little commonality. The most frequently asked questions have to do with the availability of IVR and Web-based registration and with prerequisite checking. The questions that institutions ask vendors may be closely related to the hot issues on campus at the time the document is created. Nearly one third of the questions in the registration section of Rhode Island’s RFP related to holds, for example. They may also be related to problems in the systems being replaced. The University of Minnesota’s RFP, for example, asked about the system’s ability to support term-specific registration holds. (I know from long experience there the headaches that problem can create for students and the Registrar’s Office.)

It is a bit difficult to deduce from the documents alone, without real knowledge of the institutions’ existing
processes, but it is my impression that some of the registration questions reflect a desire to do business at least a bit differently. These questions are related to things like automatic notification of students when course information, such as meeting time and location, changes, warning students when registration changes might affect financial aid eligibility, etc. This is more likely to be seen in the RFPs following the highly detailed model.

Towards Another Approach

It seems to me that the long, highly detailed functionality sections are, at least in part, and, perhaps, unintentionally, aimed at preserving the old ways of doing business. Being new at UConn, I don’t have much of a stake in doing business in the same old way. If we are going to invest the resources required to replace our systems, why not aim for “customer delight”, rather than settle for “customer satisfaction”?

As a first step to the process of replacing our student information system, we created a Request for Information (RFI). A couple methods of first determining, and then communicating, “what should be” occurred to me. One was to develop scenarios, or short stories, describing the way we wanted processing to take place in the future. I began by writing a scenario about registration.

It’s nearly time for Spring registration to begin.

Jonathan receives an email message (that’s the way he prefers to hear from the University) giving him his scheduled registration time, warning him of any holds or bars that would block his registration, reminding him of the annual registration plan he created the previous Spring, and telling him that while seeing his advisor before registering is optional, it’s not a bad idea.

When Jonathan uses the web registration system—he also could have registered via touchtone telephone, Virtual Student Services Center kiosk, or come in person to the Student Services Center—he finds a suggested registration based on his registration plan, times he had blocked out on his schedule for work and other commitments, and section availability. Where no sections are available, Jonathan is informed of alternative courses that would meet the same degree requirements and do not conflict with his schedule. Jonathan is also asked if he wants to be put on a waiting list, and request permission to overenroll. Jonathan chooses that option and also picks an alternative course. If the faculty member notifies Jonathan’s request to overenroll via email, approves the overenrollment, Jonathan will be placed in the course, dropped from the alternative, and notified of the change. Jonathan is also reminded of the number of credits he indicated to Financial Aid that he would be taking. If he registers for a different number, he will be informed of the implications for his financial aid award.

After completing his Spring registration, Jonathan enters his tentative registration plan for the coming year, using his online degree audit as a guide. A copy of the registration plan is sent to his advisor via email. The deans and academic department heads will receive aggregate information from all of the plans and the degree audits for use in the process of planning future course offerings.

Jonathan’s registration is immediately reflected in his Student Account and Financial Aid and other student system records. If he changes his registration, those changes will also be immediately reflected. As part of the registration process, or at any time afterward, Jonathan can view his student account status.

Prior to the start of Spring semester, Jonathan’s instructors receive electronic class lists, which include the students’ email addresses. Some of them send Jonathan their course syllabi and assignments for the first class session via email.

This registration scenario was included in the RFI. A group representing several student services offices on campus is now (Fall 1997) developing further scenarios, some of which will be included in UConn’s RFP. They will also serve as discussion pieces in the redesign of the student services processes here.

While writing the registration scenario I thought a lot about what characteristics a system to support that process might have and began to develop a list, another way of getting at “what should be”. Discussions with other staff at UConn lengthened the list. A review by a number of people involved in information technology or student services refined it. The following chart was the result.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
</table>

3
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-functioned</td>
<td>Includes modules to support a very wide range of student recruitment and telecounseling, admissions, transfer credit evaluation/course equivalency, financial aid and scholarship administration, student records, registration, degree audit, student accounts receivable, residential life, et cetera, transactions for undergraduate, graduate, professional, and non-degree students.</td>
</tr>
<tr>
<td>Integrated across functional areas</td>
<td>Data used in multiple modules is stored once. A transaction in one module that has an effect on data used by other modules initiates the appropriate transactions for those modules. (For example, a change in a student registration will trigger changes in student accounts receivable and financial aid.)</td>
</tr>
<tr>
<td>Meets regulatory requirements</td>
<td>Meets all applicable Federal and State regulatory requirements for financial aid and other modules, in terms of transaction processing, audit trail, and reporting.</td>
</tr>
<tr>
<td>Intuitive to use for casual users</td>
<td>Can be used by casual users, including students and faculty, without extensive training. &quot;Point and click.&quot;</td>
</tr>
<tr>
<td>Web-enabled</td>
<td>Web interfaces available for students, faculty, and advisors. Web forms should be integrated into the system.</td>
</tr>
<tr>
<td>Supports several different methods of interaction by students</td>
<td>Supports several different methods of interaction by students, including IVR, kiosk, and web.</td>
</tr>
<tr>
<td>User-maintainable</td>
<td>Highly rule-based and table-driven. Rules/tables can be maintained by non-programmers in the user areas. Allows for multiple user-defined fields, processing flags, production codes, and letter content, selection, and production.</td>
</tr>
<tr>
<td>Supports planning functions</td>
<td>Capable of a variety of “what if” projections for a range of users (students, advisors, academic department heads, and administrators).</td>
</tr>
<tr>
<td>Supports very large data files and hundreds of concurrent users</td>
<td>Supports very large data files and hundreds of concurrent users.</td>
</tr>
<tr>
<td>Year 2000 compliant</td>
<td>All modules and transactions that are part of the system are capable of functioning with dates of January 1, 2000 and beyond, as well as with dates prior to 1/1/2000.</td>
</tr>
<tr>
<td>Allows easy retrieval of information, both individual and aggregate</td>
<td>Indexed to allow fast, easy retrieval of information by student, class, account, et cetera. Supports data warehouse. Capable of a wide range of reporting functionality by numerous departments and individuals, including the ability to create customized reports, real-time reports, create ad hoc queries, et cetera.</td>
</tr>
<tr>
<td>Flexible</td>
<td>Supports multiple ways of doing business and processing rules that can be tailored without special programming to meet UConn’s business needs.</td>
</tr>
<tr>
<td>Easily extendible</td>
<td>Functions that are not supported by the base software can be created by UConn staff and easily integrated into the system.</td>
</tr>
<tr>
<td>Supports both on-line and batch transaction processing.</td>
<td>Transaction processing should primarily be on-line, but processes such as tuition and fee assessment should be capable of being run in batch.</td>
</tr>
<tr>
<td>Supports IVR</td>
<td>Supports touchtone telephone registration and access to grades, financial aid and payment status, admissions application status, and other student information.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Workflow-capable</td>
<td>Supports the routing of transactions for designated approvals and processing the transactions and generating the appropriate notifications once the approvals are given. For example, a student attempts to add a class after the 10th class day, the system would route the request to the instructor for approval; once that approval was granted, the request would be routed to the appropriate dean’s office for approval; once that request was granted the student and instructor would be notified and the transaction processed. Supports “electronic signatures”.</td>
</tr>
<tr>
<td>Supports batch upload of information</td>
<td>System allows for batch updates from various sources including tape loads of prospective student information, transcript information received via SPEEDE/ExPRESS, and upload of information such as electronic applications for admission from desktop workstations, etc.</td>
</tr>
<tr>
<td>Efficient data entry</td>
<td>Allows users to enter data into the system efficiently. Includes user field defaults and minimizes the number of screens that a user would have to enter data on for typical processes, such as applicant record creation.</td>
</tr>
<tr>
<td>EDI-capable</td>
<td>Supports electronic data interchange of information, including transcript information via SPEEDE/ExPRESS.</td>
</tr>
<tr>
<td>Paperless (or “paper-lite”)</td>
<td>Users have control over whether documents and reports are printed or produced in electronic form and over whether reports are printed centrally or remotely.</td>
</tr>
<tr>
<td>Stores user preferences</td>
<td>Capable of keeping track of user preferences for such things and receiving information via e-mail rather than on paper and also storing default values by user.</td>
</tr>
<tr>
<td>Robust security</td>
<td>Highly developed and flexible security system capable of distinguishing between update, browse, and no-access at the field level. Capable of employing data encryption. Provides for security in the electronic transmission of information to and from external entities.</td>
</tr>
<tr>
<td>Scaleable</td>
<td>System is easily re-configured to add additional concurrent users or to expand data files.</td>
</tr>
<tr>
<td>Platform, operating system, and DBMS independent</td>
<td>Capable of running on a variety of platforms and operating systems and using a variety of relational data base management systems. ODBC compliant.</td>
</tr>
<tr>
<td>Client/server architecture</td>
<td>Employs client/server architecture. Supports “thin clients”.</td>
</tr>
<tr>
<td>Object-oriented</td>
<td>Capable of storing objects, such as scanned images and digital photographs.</td>
</tr>
</tbody>
</table>

Between the scenario and the chart of desirable system characteristics, I thought we would be giving vendors a fairly good idea what we were about. I did not include information on the University of Connecticut; that information is available on the web and a serious vendor would be able to find it. I did worry that vendor responses to the scenario and the list of system characteristics might not give us all the information we needed to make an informed decision. Starting from a list of questions that the University of Maryland used in vendor evaluation, I developed, with the advice of several UConn staff, a questionnaire.

What is your corporate strategy and commitment to staying current with emerging technology? Describe current and future development initiatives.

What is your commitment to staying current with new releases of the DBMS and development tools and what level of support can we expect with upgrades?
What partnerships and relationships do you have with third-party vendors that are part of the packaged student information system solution?

Does your technology approach address a variety of customer types and scales (i.e., widespread student population vs. localized office staff)?

Besides the student information system, what other systems (e.g., finance, human resources, alumni) do you produce? Are these other systems fully integrated with your student information system product?

What types/level of training do you provide as part of the delivered solution? What training do you recommend for technical staff, user staff, and casual users? How extensive is the on-line help capability of the system?

Is there a users’ group and how frequently do they meet? What influence do clients have on new features and functionality? Describe the process that is used to address enhancement priorities.

Please list institutions with live implementations and/or partnerships with beta implementations of the most current client/server version of your software. Include head counts, enrollments, dates, scope and breadth of implementation and version number.

How do you scale your system across varying sized institutions?

What kinds of platforms is your system certified to run on, and what is involved to move from one platform to another? What is the minimum configuration and platform type that the system supports? What is your recommended configuration for user workstations?

Do you support both 2-tier and 3-tier client/server architecture? Are performance differences between the two using your software? Do you recommend one approach over the other?

What tools and methodologies do you provide to analyze performance dynamically and real-time?

What is your solution to processes that require batch processing? Can we process any transaction set we choose in batch, including transactions which are typically processed on-line? How is 7 x 24 access supported?

How are software modifications distributed to client workstations?

Is the system rules based and table driven so that modifications can be easily made to the system? What percentage of the rules are delivered as part of the base product and what percentage must be built by the customer? In a typical installation, what is the timeframe required to create the initial rules and how much effort is involved in maintaining the rules?

How much can be changed in the system via the rules table versus the program code and data base? What audit trails are provided to show who is making changes to the rules? Are the rules code driven or table driven? Are the rules sensitive to dates? What Boolean logic is used in the rules and what kind of conditions can be used in setting up the rules? Is there Cross Field and Cross Table capability? How can the rules table accommodate multi-campus policies for students enrolling in multiple campuses or joint-campus programs?

Does the system incorporate workflow management such that event notification and enforcement of the events can take place? If not, does the system integrate with third party workflow software?

Can the software make use of various E-mail packages to notify customers of certain events and action items? Does the system support MAPI for the integration of the various institutions’ mail system into the workflow of users, including automatic task execution, scheduled events and exception notification? What third party mail packages can be integrated with the system?

What is your approach to ad hoc reporting? Does your software include data warehouse functionality? How is this supported?

What is your approach to security? How well will that approach integrate with our current ACF2 software?

Is there value based security within the system that will allow for decentralization of processing, e.g., limiting access to records to department chairs and secretaries based on the value of a given field?

What audit capabilities and reports are available? Are they available for DBMS, application or both?
How is backup and recovery handled? Are these functions part of the application or of the DBMS?

What type of edit checking/validation does the system provide?

What is your solution or recommendation for document imaging? Can the system produce output that can be archived easily and can it handle the posting of input into the system from an imaging system?

What is your approach to EDI? Does it employ Internet protocols for transmission?

What processes and tools are available to assist in migrating legacy data? To facilitate incremental implementations, does your conversion process include road maps that identify the dependencies and sequences in which the various conversion modules should be run? Will these conversion processes be able to handle special data validations and business rules that institutions may wish to enforce in the migration?

Is there an archiving function and is it Rules-based? Once data has been archived, what procedures are needed to facilitate retrieval?

What is the source and presentation of documentation and help? Is the Documentation and Help customizable? Is the documentation and help field sensitive?

Does the system provide data dictionary and database architecture documentation? Is technical documentation available that depicts the physical and logical data models?

What are your estimated time frames for system implementation, with and without minimal to radical customization? Describe the level of effort needed by the institution's technical and end user staff. What are your recommendations for staffing for ongoing support?

What tool sets does the system depend on? Which of them are proprietary, third-party, or both? Which tools do you provide/recommend to in-house programmers to utilize RAD/JAD concepts while we customize your code? Are these tools available to us as part of your solution to use on development efforts unrelated to your code? Are these tools forward compatible? Backward compatible?

What is your approach to customer modification of your software? What tools or methodology do you provide to assist us in upgrading to a current release and reapplying customized changes we made to the previous version? What tools or procedures are used to provide software change management, including check-in and check-out of modules?

Can user modifications be “plugins” that don’t have to be reapplied?

Describe the steps necessary for an institution to add a field to a database table and add a field to the screen. Include impact analysis of other programs that are accessing that table.

How often are new releases made available? Is it necessary to install tool or general software upgrades before installing student information system upgrades?

Describe the type or level of support that is available for new upgrades and releases, and online, hotline support? For how long do you continue to support earlier versions of your software?

Do you have a recommended project management/system implementation methodology? What implementation support is available through your company directly? Through third-party consulting firms with partnership arrangements with your company?

How many systems have you sold to Universities of an equivalent student base, within the last five years?

Please state the universities, their technical contacts, addresses, and phone numbers.
The other pieces that the RFI included were a very short introduction and, again very short, partial listings of potential functions to be included and clients to be served. Given the information in the other parts of the document, these latter sections were probably superfluous.

While we are, at this time, evaluating responses to the RFI, work is underway on an RFP. It is likely that it will be similar in form to the RFI. The introductory section will be somewhat different and there will be additional scenarios, dealing with areas outside of registration. The questionnaire section will include questions on pricing, implementation support, and training. As a document, it will be quite different from the other RFPs I have reviewed.

**Some Conclusions**

When I sent my requests for copies of RFPs, I also asked for impressions of the effectiveness of various models. The responses, regardless of the approach taken to creating an RFP, seemed quite similar. The purchasing decisions were based on more than just the information included in the RFP responses. Vendor visits to campus, site visits to other institutions, and discussions with colleagues at other institutions all played important roles in the process. Regardless of how the decisions were made, and what vendor was selected, there were surprises during implementation process. It may be that looking at the vendors’ customer lists and deciding based on which has institutions most closely matching yours would work as well as anything else.

That being the case, why choose one approach over another? One important advantage of the scenario-based approach is that it forces you to think, at least at a high level, about how you want to do business in the future. Even if, or even though, the vendor solution will not accommodate all of the features of your scenario, you will have begun the process of considering how the system will function when implemented and you may begin to address process change issues sooner. Scenarios also may be of assistance in “selling” the project to those who have to approve funding, as it may help them understand how the very large expenditure of funds and effort will result in transformed services. Later on, they may supply a kind of high level road map for those involved in the implementation phase of the project.