Education Management Information Systems (EMIS):
Available Software and Guidelines for Selection

Kurt Moses, Vice President
Vivian Toro, Director
Academy for Educational Development

Selecting EMIS software (whether custom designed or prewritten) in its details is complicated, but the key outcomes are not complicated —making existing jobs work better: more accurately, more timely, and more efficiently. Moreover, good, sustainable EMIS efforts in the public sector are evolutionary, not revolutionary—they build as much as possible on what already works well. Software (not hardware) should be the focus of attention of a Ministry or its agencies, since software selection for EMIS should drive whatever additional hardware is needed.

Strategic Choices

There are several important distinctions to know about EMIS software before deciding what to select.

Transactional versus Management or Planning Software. In the early stages of EMIS most Ministries think of EMIS as an undifferentiated set of solutions to all their problems involving computers, reporting, data, and costs. A key distinction for large educational systems is between transactional software and planning-oriented or reporting software. Transactional software performs the daily, individual functions such as payroll, tracking personnel, recording leave, tracking inventory of books, or accounting for payments. Such software tends to be highly specialized, accommodates considerable detail and can be used to derive reports that are used for management and planning. Some software is sold as payroll only, with no personnel function; other is integrated using a common database and common procedures and computer screens. If an educational system decentralizes and places responsibility for recruiting, tracking, and paying teachers at the District level, for example, then the District needs, at a minimum, transactional software for payroll and personnel. Typically, up until decentralization, most payroll software will be at a central ministry level. **Hence, when large educational systems start moving responsibility for certain key functions—such as payroll, personnel recruiting and tracking, financial accounting, and tracking vehicles to a lower level, they must also plan on software and hardware support for these functions.** What many governments do when they decentralize is to force Districts or lower levels to use manual procedures even when, at the central level, most things are automated.

Developing Versus Buying Software. Until about 10 to 15 years ago, the only major option was to make your own software—custom designing it for a particular function, such as payroll, and operating on very specific equipment—often mainframe computers, with very specific computer languages—such as COBOL or Neat3 or Fortran. That era has been far surpassed. With more advanced software, the use of Internet languages like HTML or Java, and the widespread use of PC computers, there is a very large selection of prewritten software for almost any administrative function or any sized organization, that operate on most internationally compatible PCs, servers, or minicomputers. As well, the educational market has gotten big enough that specialized software firms now provide both individual and integrated software for schools and Districts (or regions) that support most of the functions that an independent or fully decentralized school might need to perform. Additionally, many software vendors now provide government oriented software for large volumes of transaction such as payroll for 1 million staff, or personnel tracking for 2.5 million administrative staff. In short, an effective Ministry can now find, proven, tested, documented, and easy-to-train-for software for almost any need at a reasonable price. **The key advantage with prewritten software is that you can try it out before you buy it, it goes in much faster than custom software, and a vendor can provide support afterwards should problems arise. In most instances, prewritten software, which is then customized to special needs, is the only time-effective means to get a nation-wide EMIS working in a reasonable period of time.** In some developed countries, software vendors are now offering remote access to their software on a per month or per transaction basis.

Some key things to look for with prewritten software:

- History of successful use in situations like yours—at least 3-5 years;
- Ability to be used in different sized or volume situations, such as district, regional, province, and national(scalability);

- User friendly—people can operate it quickly with minimal additional training;
- Internationally compatible databases and procedures (there are a series of widely accepted international standards and databases available);
- Supported locally by a reputable vendor—there is a financially solvent vendor around to resolve issues;
- Multiple site licenses—it should not cost a huge amount of money to add an additional site;
- Transparent pricing—the Ministry should be able to evaluate the cost of purchasing and operating the software for 5 years, easily. Pricing should be done according to a "life cycle" software costing approach;
- Web compatible—at some point, the software should be accessible via the Web either for reporting or data gathering purposes;
- Excellent training—there should be excellent and continuous training available for at least 5 years;
- The vendor should be able to demonstrate and explain to any management person in the Ministry what the software can do for them. Strictly technical statements are NOT sufficient; and,
- Give yourself at least 6-8 months to do this work with a small team.

Sample List of EMIS Software

Exhibit A is a sample list of single function as well as integrated education-oriented management software. The list is indicative and not exhaustive. As you review it, keep in mind that software suitability is highly dependent upon the local situation, particularly in terms of functions, support, training and cost.

Steps to Software Selection

Step #1 – Conduct a Needs Assessment

What kinds of information do you need? Who will use the information? For what purpose? To answer what questions? How frequently do you need the information? What links does it have to existing or planned information systems inside and outside your organization? How does it fit into your institution's overall EMIS plan/vision? Does your organization have (or can afford) the necessary pool of skills to develop, customize, install, operate and/or upgrade the necessary software? Do you need to consider contracting, outsourcing or training existing staff? These are some of the questions you should try to answer during this step in order to get a realistic perspective.

The tendency in developing or selecting EMIS systems is to try to use a Statistical Information System (SIS) to meet the information needs that are best met by transactions-based administrative systems such as Human Resources/Personnel, Financial, and other transaction-based systems. But this will only be a short-term solution. At the very minimum any national education system should address basic educational questions such as:

- How many schools do you have?
- Where are these schools?
- How many children/learners are in the education system?
- What are the basic characteristics of these children/learners?
- How many teachers are there?
- Where are they?
- What is the basic infrastructure of your schools?
- What is the basic financial information?
- What basic teaching materials does each school have?

The needs assessment should be done in a participatory fashion, consulting as many stakeholders and users as possible. The assessment process should take into account what works well in current systems whether they are manual or automated processes and functions, and build on these successes. It should also look at the "must-have" features, because not everything needs to or can be done at once.

Common pitfalls:

- Defining systems as only automated systems. People have manual or partly automated "systems" that work and should be included as part of a needs assessment.
- Leaving the needs assessment only to the "technical" people, rather than including managers from the beginning.
- Excluding your "technical" people from the needs assessment process.
- Looking at software acquisition as a one-time purchase and not as a system-wide investment.

Step #2 – Specify System (Software) Requirements

In order to obtain the kind of information system you want, you must be as clear and specific as possible about what kind of functionality you want from a software system(s) and vendor. Some of the factors to consider in specifying software requirements include: ability to perform key functions, compatibility or interlinking with available software and environment; cost (core system, initial conversion, training, support, upgrades, customization, and maintenance); user friendliness; adequacy of documentation; access to technical support; open source code or proprietary software, sustainability, and upgrades.

Common pitfalls:

- Not developing system requirements.
- Not separating critical system requirements from non-critical requirements.

Exhibit A: Sample List of EMIS Software

Name and Contact	Area	Sample Functions	Country of Origin	Comments
Education Management Systems www.ems-isis.com	Integrated School & District Software	Finance, School Lunches, Student Records,	USA	Small, medium schools & Districts
Class Act Software Info@classactsoftware.com	Integrated School Software	Student Records, Teacher Pay, Finances, Class Attendance	USA	Small, medium sized schools
TASS—Alpha School System www.alphabus.com.au/tass/tass.html	Integrated School Software	Student Records, Teacher Pay, Finances, Class Attendance, Student Accounts	Australia	Used primarily in Australia, variable school sizes
Powerschool www.Powerschool.com	School Student Software	Student Records, Class Attendance, Parent Contact	USA	Supports instructional activity best
ABT Campus www.abtcampus.com	Integrated School Software	Student Records, Class Attendance, Business Management	USA	Extensive use of the Web for interfaces
Rediker Software www.rediker.com	Integrated School Software	Student Records, Class Attendance, Counseling Records, Business Management	USA/ Europe	Worldwide application, oriented to educators needs
SchoolPro www.schoolpro.com	Integrated School Software	Student Admissions, Records, Billing, Business Management, Payroll, Facilities Management	USA	
IBM—Solutions for Schools www.ibm.com/solutions/	School Software, Selected District Software	Various	USA/ Worldwide	Various semi-custom solutions
Computer Associates www.ca.com/products	District, Regional, National, Single Function Software	Finance/Accounting, Human Resources, Inventory	USA/ European	Requires a Systems Integrator to link large systems
MSA Inc. www.msa.com	National, Provincial, Single Function Software	Human Resources, Inventory, Textbook Management	USA/ Worldwide	Requires Systems Integrator to link large systems
PeopleSoft, Inc. www.peoplesoft.com	National, Provincial Semi-Integrated Software	Human Resources, Financial, Student Records	USA/ Europe	Requires Systems Integrator to install.
ED*ASSIST www.aed.org./edassist	National, Provincial, Regional, District Integrated Software	Information reporting on Students, Human Resources, Financial Summaries, Facilities, Textbooks	USA	Planning & management oriented EMIS. Links to existing school system.
SCT Solutions www.sctcorp.com	National , School Integrated Software—also Higher Education	Students, Human Resources, Finances, Inventory, Class Scheduling	USA/ Europe	Higher Education oriented. Large user base.
Campus America www.campus.com	School Integrated Software—Higher Education	Student Records, Human Resources, Finances, Inventory, Class Attendance, Store Management	USA	Higher Education oriented. Medium sized institutions

Step #3 – Develop an Evaluation Criteria

In evaluating your options, you should develop evaluation criteria that help you quantify the priorities and features that are more important to you. This process should help you to weigh how much emphasis you will give to certain technical and non-technical categories. For example some of the technical categories might include application functions, the database management system, maintenance, training, and support, while non-technical categories might include cost. These categories should be directly linked to the detailed list of system requirements developed in step #2.

Exhibit B summarizes in matrix form an example of the Software Evaluation Criteria used to acquire Human Resource Software for a Ministry of Education in Latin America for a decentralized system. In this particular situation, the ministry had decided to "buy" a software system that met about 70% of their key requirements, and to customize the remaining 30%. You should complete these evaluation criteria for each one of the software top candidates.

Common pitfalls:

- Not having systematic evaluation criteria.
- Attributing too much importance to technical features and not to the overall operation environment.
- Evaluating software as an independent activity and not as part of an overall plan.

Step #4 – Identify Existing Software Options

Once you have specified your system requirements (or as part of that process), you should try to identify existing software options. Should you (or a contractor) develop a totally customized software system or do you acquire existing "off-the-shelf" software?

In order to determine what is the best solution to meet your needs, you need to see if software solutions exist, and, if they do, do they meet all, or some of the needs. You need to determine how much time, effort, and resources would be needed to customize an existing system, or whether it is better to develop your own system "from scratch." The latter is always time-consuming, resource intensive, costly and frequently hard to sustain. The answer might be a compromise between the "make" and the "buy" options since there are benefits and constraints to both options. The solution will need to take into account your information needs, your institutional capacity, and existing resources.

Common pitfalls:

- Wanting to "invent," believing that local development is cheaper.
- Acquiring/developing software that answers current questions but is less flexible at adapting to the inevitable changes in policies and priorities.

Step #5 – Obtain Feedback from Users

In addition to the research on software advertisements, Web sites, demonstration programs, vendor presentations, professional publications, and software reviews you should also try to obtain feedback from "real" users. Consult with other colleagues or collaborators trying to answer the same questions, and having the same problems.

In particular, you should talk to similar organizations and ministries that have similar information needs. Find out about lessons learned. What do they like about the software? What are the shortcomings? If they were doing it all over again, would they use the same software? The same vendor? The same contractors/consultants? Would they do the development "in-house"? Would they outsource it? Would they do a combination of both? Are the users happy? What kind of infrastructure is needed to support the software? What is the "real" cost? Can the vendors/consultants show you a prototype of how the system would look once adapted to your needs?

Common pitfalls:

- Relying only on vendor "claims" about their software.
- Relying on software that is not really oriented to your "activity."

Step #6 – Consider Acquisition Process and Preconditions

You should become familiar with the acquisition procedures within your organization and, if applicable, those of the funding source. Each institution and each donor has different procedures, limits, formats, and cycles, that could significantly delay your acquisition. You should also synchronize this process with other procurement procedures that are preconditions or requirements for the successful implementation of your system. This might include hardware and software procurement and installation, as well as staff training ranging from Office Automation (word processor, spreadsheets, Internet use, etc.) to different levels of specialized technical training (databases, operating systems, networks, etc.). It could also include the readiness of the different user groups, particularly if you plan to install in regional or District offices.

Common pitfalls:

- Not knowing the donor's or your own software procurement procedures.
- Not synchronizing your software acquisition with other relevant aspects.

Step #7 – Develop Feedback Mechanisms

Feedback mechanisms are critical throughout the whole process of procurement, customization, installation, refinement, training, and maintenance -- particularly from users of the system. Use routine feedback mechanisms such as evaluation forms, or group interviews, and create an environment in which users feel their voices are "heard" and taken into account. Also maintain the lines of communication with the vendor/consultant/contractor so that there is some flexibility incorporating feedback. Additionally, invest in a service contract that provides for "hot line" support, system upgrades, access to user groups, etc.

Finally, always develop contingency plans, just in case things don't work out as expected. People leave, companies go out of business, technology changes, systems become obsolete.

Common pitfalls:

- Assume that the acquisition process stops once a software selection has been made.
- Not having a contingency plan.

Software is just part of the process of building your EMIS. One of the measures of success will be when users start demanding more from the system because their original expectations are being met and they expect more. When new systems are introduced, staff expectations may rapidly move from expecting a 2- month delay to get a new teacher paid, to expecting 1 week. Today's questions will quickly be replaced with new ones. Accordingly, a potentially successful EMIS should have a built-in mechanism for evolution and growth.

Exhibit B: Software Evaluation Criteria

MAIN CATEGORIES	SPECIFIC CRITERIA	MAXIMUM VALUE	VALUE FOR CANDIDATE A	VALUE FOR CANDIDATE B	VALUE FOR CANDIDATE C
General Characteristics	Company profile	5			
	Software Package General Characteristics	2			
	Software Architecture	5			
	System Internals	5			
	Customization/Adaptation	5			
	Distributed Processing Support (Network)	5			
	Licensing	3			
	Cost	30			
	Initial	15			
	Recurrent	10			
	Customization	5			
	Support	10			
	Training	8			
	Documentation	5			
Total - General Characteristics		83			
Software Characteristics	Recruiting/Applicant Tracking	1			
	Post Assignment Proposals	1			
	Employee Information	3			
	Positions	2			
	Personnel Actions Tracking	1			
	Reporting	2			
	Ease of Use	2			
	Data Volume Handling	2			
	System Interface	3			
Total- Software Characteristics		17			
Total Evaluation Criteria		100			