Cost-Justifying Electronic Performance Support Systems

How can the benefits and return on investment of an EPSS be determined in advance?

Consider how attractive and valuable a support system would be if it allowed any individual to enter a new job without prior training, and if it would gradually bring this individual to higher performance levels than those achieved by traditional initial training sessions. Not only would such a system offer savings in training costs, but it would also provide productivity gains due to the user's increase in performance—the ultimate aim of Electronic Performance Support Systems. How realistic is this goal, and how cost-effective is an EPSS in reality? In spite of a number of success stories,[1] this question is difficult to answer with certainty, especially in a pre-project phase when the intent is to determine if an organization should go ahead with an EPSS project. The large number of factors to consider and the lack of data to help predict results make it difficult to quantify the return on investment for an EPSS. The objective this article is to provide some empirical data on this subject by using cost-benefit data based on the analysis of a potential EPSS project for a large electric utility company. We hope the findings will give some generalizable results and that the method can be reused for the purpose of conducting EPSS cost-benefit analyses.

Michel C. Desmarais, Richard Leclair, Jean-Yves Fiset, and Hichem Talbi

[1]A number of case studies are compiled at http://www.epss.com/lb/lb_index.htm. They cite, for example, a study reporting that the use of an EPSS for customer service employees reduced training time by 60%. Another study of the use of an EPSS for a front-end interface of a bank legacy system reports a reduction in on-the-job training by 75%, a reduction of time per task from 33% to 77%, and a similar reduction of errors.

Characteristics of an EPSS

The fundamental goal of an EPSS is to provide assistance in learning and in performing some set of tasks. Some authors go so far as defining the role of an EPSS as an "electronic infrastructure that captures, stores and distributes individual and corporate knowledge assets throughout an organization" [7]. With such broad objectives, it is not surprising that the term EPSS encompasses a large breadth of systems and that there are many definitions of an EPSS [1, 3].

In an enlarged definition of EPSS, the target tasks to learn and execute need not be performed through a computer application. In this case, EPSS are closer to the notion of computerized help desks in which the sole purpose of the computer is to assist someone in performing some task, such as the diagnosis of a machine failure or the evaluation of an individual's personal credit. EPSS that are embedded in a computer application often take the form of well-known features such as wizards, cue cards, or other advanced help features.

Another important distinction should also be made when the goal of the EPSS is to provide assistance in performing a task as opposed to assistance in learning the task.[2]

For the purpose of this study, we limit our scope to EPSS embedded in computer applications and to features that provide assistance in doing as well as learning a task, and not solely assistance in doing the task. Nevertheless, we presume many of the findings here might also apply to a standalone EPSS.

When the conditions are favorable, EPSS can provide huge benefits in a variety of areas:

- *Enhanced productivity:* Often the most significant returns will come from enhanced worker productivity stemming from just-in-time support and continuous learning. In addition to performance support, an EPSS can offer a rich learning environment allowing the user easy access to useful information that otherwise would never be consulted. This benefit is most significant for occasional users who find it hard to remember all the details about their tasks. Job rotation and seasonal tasks are good examples of context in which EPSS can be very beneficial.
- *Reduced training costs:* The availability of an EPSS on the job can reduce the initial training phase to the minimum set of skills to perform the job. Workers learn the rest of the skills necessary for good performance while using the EPSS to do the job.
- *Increased worker autonomy:* EPSS provide an information-rich environment in which the individual is not only better supported to perform his/her job, but can simultaneously acquire the knowledge to improve, thus reducing the burden on support teams and allowing for more worker autonomy.
- *Increased quality due to uniform work practices:* One consequence of providing uniform information and procedures to all workers by means of an EPSS is the reduction of variability in work practices. This constitutes a favorable outcome in many contexts, for example, in a customer service department: The customer is systematically given the same answer to the same problem, no matter who answers the call.
- *Knowledge capitalization:* Designing an EPSS generally involves the expertise of experienced employees and formalizing the system for easy access. It also allows for the continuous addition of useful information by employees. Consequently, EPSS are a means of documenting and formalizing the knowledge capital of an organization [7].

A number of companies have claimed large gains from the implementation of EPSS within their organizations. Among them is the American Express customer service department, which claims that the training period for their employees was reduced from 12 hours to 2 hours, that productivity improved from 17 minutes per request to 4 minutes per request, and that the data entry error rate decreased from 20% to 2%.[3]

This article focuses mainly on the benefits of reduced training costs, as the empirical study and cost-benefit analysis were conducted within the perspective of a training department in a large utility company. However, the reader should keep in mind that this is only one of many potential benefits. In spite of a significant number of convincing success stories, EPSS are not necessarily a sure win. The conditions of success must be met, and they are numerous. The success factors are both technical

[2]As an example, consider a problem one of us encountered in writing this text on a PC: The computer lost track of the CD-ROM driver for an unknown reason, and the user proceeded with the wizard dialogue designed to "recognize" the installed hardware on the PC. The wizard did "find" the CD-ROM, but the mouse stopped responding! Although the wizard helped the user in performing the task of reinstalling the CD-ROM driver, it provided no insights into the "manual" way of installing/de-installing hardware components. Thus, when confronted with the problem of fixing the mouse driver installation, the user had no help from the wizard—neither in performing the task (in fact the wizard itself triggered the problem), or in learning how to do it.

[3]Reported in the case studies at http://www.epss.com along with a number of other success stories.

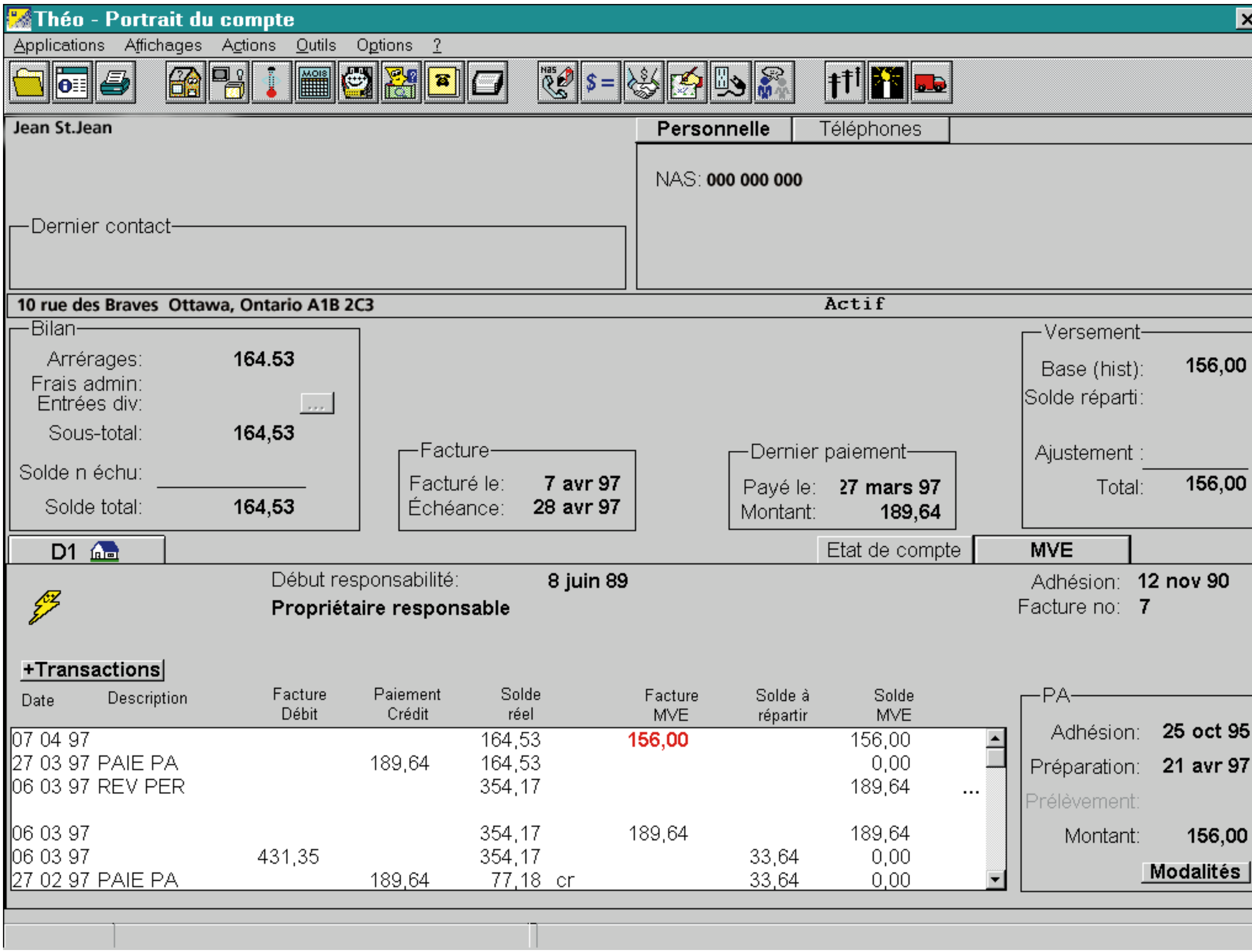


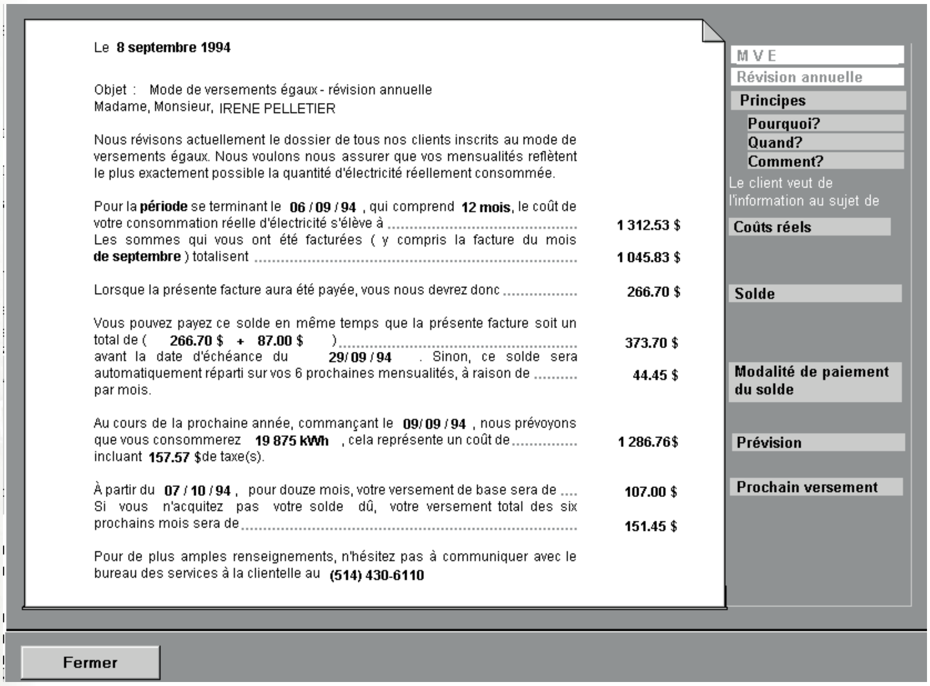
Figure 1. The standard THEO interface

and organizational. They include, for example, the number of users, the availability of domain experts and EPSS implementation specialists, and the amenability of the tasks themselves to this type of learning support. Even when the conditions for success are present, the uncertainty of outcome for an EPSS project might still be too high for management to justify the project.

It is with regard to such uncertainty in a potential EPSS project that we conducted an empirical study around a cost-benefit analysis. The aim of this study was to identify the most sensitive success factors for the EPSS project, assess their impact on project outcome, and assess the overall project's return on investment for different scenarios. Most importantly, the study involved an experimental phase in which a prototype of the EPSS was implemented and tested with users. The prototype thoroughly covered a subset of the users' tasks the projected EPSS would need to cover. We believe reliable data could only be obtained from a realistic analysis that closely replicates the operating environment, even if this analysis covers a subset of the users' tasks, and that the findings should extrapolate reasonably well to the overall system.

The THEO-EPSS Prototype

THEO is a customer service application used by one of the world's seven largest utility companies. This application can be thought of as a graphical user interface to the company's customer database (see Figure 1). The customer service representative (CSR) receives telephone calls from customers and potentially has to perform a number of tasks with THEO, such as creating a new customer account, modifying an existing account following a change of address, and providing explanations about the customer's electricity bill. One of the most difficult and frequent tasks for the representative is to explain the Equal Payment Plan, a program allowing customers to pay equal monthly amounts instead of amounts based on the previous month's electricity consumption (which fluctuates considerably from winter to summer months). This is the specific task addressed



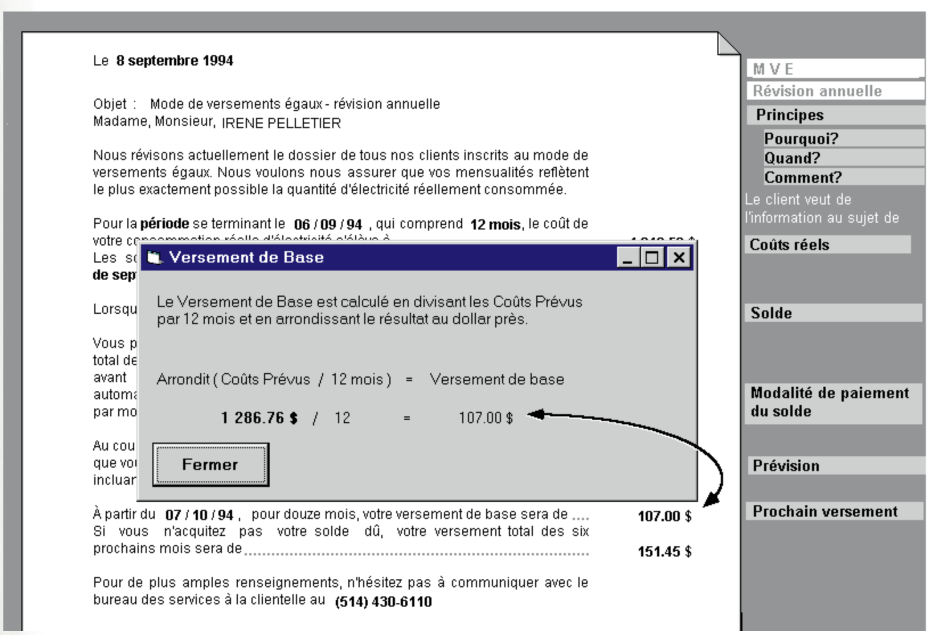


Figure 2. (top) The THEO-EPSS interface
Figure 3. (bottom) Example of a number decomposition that explains how the basic payment value is created

by the THEO-EPSS prototype.

Two different techniques were used for the purpose of this experiment. They are both implemented as functionality embedded in a new screen that mimics the letter sent to the customer by the utility company (see Figures 2 and 3). This screen displays much less information about the customer than the standard screen. Moreover, it contains a large amount of text that is the same for all customers. However, it allows a common point of contact between the customer and the CSR that facilitates communication. There are two techniques available to a CSR using THEO-EPSS to support their task of explaining the electricity bill to customers.

- *Number inspection technique:* The first support technique consists of allowing the user to "inspect" a number that appears on the computer screen by clicking on this number (see Figure 3). The number acts as a hypertext link and displays the details of how this amount is computed. This technique is recursive such that the explanatory text can itself contain numbers that can further be inspected in the same fashion.
- *Hierarchical menu technique:* The second support technique is based on a classification of all potential customer questions into a hierarchical decision tree (the menu-like decomposition on the right-hand side of the screen). The user looks for information in this hierarchical menu.

The same information can typically be obtained using either of these two techniques.

Experimental Procedure and Subjects

The intent of the experiment with the THEO-EPSS prototype was to provide enough data, both qualitative and quantitative, to assess the ability of the EPSS to reduce initial learning and increase user performance. The results from this experiment subsequently served as input to the cost-benefit analysis, which is based on three scenarios: best-case, expected, and worst-case.

The experiment was not intended to be a scientifically rigorous proof of the functional objectives of the EPSS. This would have required a more systematic procedure and a larger sample size. It serves solely as evidence for the cost-benefit analysis to delineate estimated margin of error.

To determine a realistic set of tasks to perform with the prototype, we analyzed all questions and requests from 31 pre-recorded phone calls to the customer service department concerning the Equal Payment Plan. From this protocol data, 15 tasks were derived that covered all the 31 sample requests. It is thus realistic to assume that these tasks cover most, if not all, the usual requests that a CSR would have to address.

The intent of the experiment was to provide enough data, both qualitative and quantitative, to assess the ability of the EPSS to reduce initial learning and increase user performance.

The CSR needs to find the necessary information in the THEO interface to answer the customer's query. The three following queries are examples of frequent tasks:

- "Why do I have a balance to pay?"
- "How is this balance determined?"
- "If I decide to spread the balance payment over the next months, what will be the monthly payments?"

Three experimental conditions were defined:

(1) In the first experimental condition, the 15 tasks were submitted to three novices and two experienced subjects who were asked to find the information in the THEO-EPSS interface. The interface consisted of the original THEO information complemented by the EPSS.

(2) In the second condition, one novice and three experienced subjects were asked to use the standard THEO interface without access to the EPSS. However, only the tasks for which the information could be found on the standard THEO interface were submitted, which reduced the number of tasks from 15 to 8.

(3) Finally, another experiment was conducted to find out the extent to which practice would reduce the time it takes to complete the tasks. Therefore, the same tasks were administered repeatedly. Two subjects were asked to complete the 15 tasks at 3 different times during the same day.

The novices were people with no knowledge of THEO's interface or knowledge of the overall job itself, and were not employees of the utility company. The experienced subjects were company employees who were active as customer service representatives, or who had been active recently, but who did not know the THEO and THEO-EPSS interfaces.

The experimenter kept track of the number of tasks correctly completed and the overall time it took each subject to go through all tasks, regardless of how many they successfully completed. The subjects were also asked to complete a questionnaire at the end of the experimental session. They were asked to provide feedback on the perceived usefulness of the EPSS.

Experimental Results

In the experiment's first condition, in which the EPSS was present, both of the experienced subjects and two of the three novices were able to complete all 15 tasks. The novice who did not complete all tasks was able to answer 10 of the 15 queries correctly. It took subjects between 30 and 40 minutes to complete the task, with an average time per task of 2.2 minutes.

In the experimental condition where only the standard THEO interface was available, the novice subject failed badly with only one out of the eight tasks completed. In fact, the subject commented that it was more or less impossible to answer the questions unless you knew the Equal Payment Plan. It was also fairly obvious that the other seven questions omitted from this part of the experiment would be very difficult for a novice to answer. On the other hand, the experienced subjects performed relatively well with two out of the three completing all eight tasks and one completing six out of the eight.

The experimental condition designed to measure the learning curve showed that the time to complete all 15 tasks was reduced by half (from 30 minutes to 15 minutes) from the first trial to the third. Therefore, we would expect that given a few hours of practice, a user could rapidly locate the necessary information and that the delay could be acceptable in an on-the-job context, as long as this delay does not occur more than once or twice during a telephone conversation.

Results from the qualitative questionnaire were very positive. All subjects, whether novices or experts, had positive comments on the EPSS. They all considered this tool to be useful, especially for novices. The number inspection technique was preferred over the hierarchical decomposition. It was unanimously considered "very useful" by all five of the subjects who filled out the questionnaire, whereas the hierarchical decomposition technique was considered by two subjects as "very useful" and by the three others as "rather useful."

Cost-Benefit Analysis

The experiment gave us clear evidence that the THEO-EPSS can provide useful knowledge to novice users and allow them to perform well on tasks that would normally require initial training, showing that THEO-EPSS can reduce initial training.

- It can be used during training in the form of self-practice sessions, which would have the benefit of providing knowledge in exactly the same context as the operational situation, a factor known to favor the recall of relevant knowledge.
- Secondly, the targeted level of mastery after initial training can be reduced by assuming that the lower level of skills will be compensated by the availability of this help tool on the job.
- We can further assume that the hard-to-recall information would be more efficiently transmitted to the customer if THEO-EPSS is used. This is particularly important because some customer queries are encountered rarely or only at specific periods during the year.

Do the benefits of reduced training and increased employee performance outweigh the costs of EPSS development and maintenance? We investigated this question by conducting a cost-benefit analysis based upon three scenarios:

- *Worst-case scenario:* an unfavorable situation in terms of cost or benefit unlikely to occur;
- *Expected scenario:* the most likely situation;
- *Best-case scenario:* an unlikely but favorable situation.

Because the study was commissioned by the training department, we considered only reduced training benefits in our cost-benefit analysis. This conservative stance left out other potential benefits. In particular, potential benefits stemming from overall increased employee performance (due to the continuous training nature of an EPSS) and benefits stemming from increased quality of the employee's answer to a question (due, for example, to the uniformity of the answers provided to the customer across different representatives or across time[4]), were not considered. Although these other benefits can be very significant, they are very difficult to measure or to estimate, and therefore we decided not to include them in the cost-benefit analysis. As a result, the estimated benefits from the study should be interpreted as the minimal potential figures that would be increased by consideration of the other benefits.

For each of the three scenarios (worst-case, expected, and best-case), estimates were produced for the following development cost factors:

- Implementation cost for the "number inspection" EPSS technique
- Implementation cost for the "hierarchical menu" technique
- Development cost of the learning content itself

In addition, estimates were produced for reduced training cost concerning use of the application.

Given that these estimates are relatively independent of each other, the resulting worst-case and best-case scenarios constitute true extreme estimates—the probability of co-occurrence of favorable or unfavorable situations equals the product of their individual probabilities.

Hypotheses

The cost-benefit analysis relies on a number of hypotheses that are explained briefly in the following paragraphs.

Reduction in initial training = 50% (best-case), 30% (expected), 20% (worst-case). The experimental results clearly demonstrate a performance improvement over the standard THEO interface, and a learning effect as well. A reduction of the initial training period is thus expected from these results. Two effects can account for such a reduction:

[4]For example, a negative perception can occur if a customer is given two different explanations for an increase in last year's electricity consumption by two different customer service representatives.

(1) The EPSS can be integrated into the initial training program and increase its efficiency. Several factors make the EPSS a good learning tool, namely:

- It adapts to the individual learner's pace.
- The learning context exactly matches the operational context.
- It represents an engaging learning mode in which the user has to actively seek the information needed to solve problems.

(2) The on-the-job availability of the EPSS permits the initial training program to target the minimum set of technical skills; instead of attempting to teach all of the technical skills (mainly procedural) required on the job. The training program can assume that a subset of skills will be acquired on the job through the EPSS and thereby focus only on core competencies.

Table 1. Cost-benefit analysis summary results (data is approximated)

		Scenario		
		Best-case	**Expected**	**Worst-case**
	Development cost			
a1	(1) Software: number inspection	$ 2,511	$ 4,860	$ 5,994
a2	(2) Software: hierarchical menu	$ 14,229	$ 27,540	$ 33,966
a3	(3) Software: combined (a1 + 2a)	$ 16,740	$ 32,400	$ 39,960
b	Pedagogical content	$ 4.810	$ 11,470	$ 18,130
c1	*Total cost: a1 + b*	*$ 7,321*	*$ 16,330*	*$ 24,124*
c2	*Total cost: a2 + b*	*$ 19,039*	*$ 39,010*	*$ 52,098*
c3	*Total cost: a3 + b*	*$ 21,550*	*$ 43,870*	*$ 58,090*
	Annual benefits			
d	Training time reduction	50%	35%	20%
e	Training days devoted to experiment's task set	1,5	1,5	1,5
f	Cost per participant	$ 500	$ 500	$ 500
g	Annual number of participants	60	60	60
h	Annual benefits stemming from training time reduction (d x e x f x g)	$ 22,500	$ 15,750	$ 9,000
	Other annual costs (maintenance and capital investment cost)			
i1	10% of c1	$ 732	$ 1,633	$ 2,412
i2	10% of c2	$ 1,904	$ 3,901	$ 5,210
i3	10% of c3	$ 2,155	$ 4,387	$ 5,809
m1	*Net annual benefits (h – 11)*	*$ 21,789*	*$ 14,117*	*$ 6,588*
m2	*Net annual benefits (h – 12)*	*$ 20,596*	*$ 11,849*	*$ 3,790*
m3	*Net annual benefits (h – 13)*	*$ 20,345*	*$ 11,363*	*$ 3,191*
n1	Break-even point (years), number inspection technique only (c1/m1)*	0.3y	1.2y	3.7y
n2	Break-even point (years), hierarchical menu technique only (c2/m2)*	0.9y	3.3y	13.7y
n3	Break-even point (years), both techniques combined (c3/m3)*	1.1y	3.9y	18.2y

* The values in rows n1 and n2 assume the reduction in training time is the same regardless of whether any of the two techniques is used or if they are combined.

However, the experimental results also suggest that initial training cannot be eliminated completely. Indeed, some subjects could not complete all of the 15 tasks, and more importantly, the time required to complete the tasks is considerably longer for beginners who lack both familiarity with the task and the EPSS, as compared to experienced users who have complete mastery of the task and EPSS. Thus, we considered that the EPSS tool could not replace initial training, but that it could be used with training to bring users up to an acceptable level of mastery. Experimental results showed that even if the user did not know an answer, the information could be rapidly identified within a few minutes given some preliminary practice with the EPSS.

Given these considerations, the range of initial training reduction was fixed at 50% for the best-case and 20% for the worst-case scenario, with an expected gain of 30%.

Cost of software tools development: an additional 15% of original interface development cost for the "number inspection" technique, and 85% for the "hierarchical menu" technique. These numbers were directly obtained from the software development service itself. They took into account the number of new modules that would require additional software development vs. the existing modules that could simply be reused, the number of dialog windows involved, and certain other factors that are normally considered in such a process of development cost assessment. This factor was not subject to a best/worst-case scenario as the prototype was an operational version of the required interface that left few implementation uncertainties and provided a very accurate specification of software development needs.

Pedagogical content development cost: 0.5 person-day (best-case), 1.0 person-day (expected), and 1.5 person-days (worst-case), for every information capsule developed (there were 26 in total and they correspond more or less to a dialog window). These numbers were estimated from the experiment. An information capsule consisted of a short explanation text that was generally displayed in a separate window or frame. The worst-case scenario was obtained by summing up the total time required for the initial pedagogical content development and this sum was divided by the number of information capsules (26) to yield the average development time. We assumed the very first pedagogical development time would incur trial-and-error efforts and that a significant productivity gain would be acquired through practice, and thus that this original time would constitute an upper boundary. The set of tasks we chose (customer invoice explanation) was considered one of the most difficult for the representatives, and it is likely to include the most difficult pedagogical content development as well.

The training and cost factors were estimated from data derived from the experiment. Other factors also had to be estimated, but the estimates were all based on information available within the organization. This information could all be derived from existing accounting department figures or from the training department.

For example, we estimated the number of days devoted to learning the tasks performed in the experiment session (tasks related to explaining a customer's billing procedure and computations) to be approximately 1.5 out of the total five initial training days. The estimate is based on the proportion of pedagogical content dealing specifically with the corresponding tasks and, in addition, a portion of the "core" knowledge required for all tasks.

The participant's cost is *not* the one obtained for the utility company—this cost is not disclosed here as the information is considered confidential. The recommendations for this study are thus slightly different from the actual ones due to this discrepancy. The figure of $500 per day is taken as a reasonable amount that includes both the trainer's salary and training service overhead cost, plus the participant's salary, social benefits, and travel expenses when appropriate. It excludes training department costs related to course design, strategic planning, research and development, and personnel training.

Results

The results of the cost-benefit analysis are summarized in Table 1 using approximated data. The results are divided into the three scenarios as discussed previously, namely best-case, expected, and worst-case scenarios. In addition to calculating the global EPSS development cost, we kept separate accounts of the development cost attached to each of the two individual EPSS techniques, namely the number inspection technique and the hierarchical menu technique. This separate accounting outlines the major difference in implementation cost between the two techniques. Each technique thus has a very different impact on the return on investment and that difference had to be taken into account in the recommendations made to management.

There is a strategic decision involved: the potential of this technology is great and the company could gain experience now instead of waiting for later.

The table's last line (n3) summarizes the conclusions we can draw from the cost-benefit analysis for the combined techniques. It indicates that the break-even point is likely to occur after about 4 years (3.9 years), and that it can range from 1 to 18 years. Another analysis, based on the internal rate of return over a five-year period after the development phase, would indicate a 94% rate for the optimistic scenario, 9% for the expected, and a negative rate for the pessimistic scenario.

The details show that the number inspection technique has much greater potential. The total development cost is much lower because it can rely upon a variety of existing software modules, such as WinHelp. Moreover, the hierarchical menu technique can be replicated with the number inspection technique (since both can be considered derivatives of hypertext-like dialogs). We expect that doing so would provide slightly less elegant and seamless interactions, but it would still allow easy access to the same information. Although we cannot claim that this technique alone would result in the same learning and performance increases as both techniques combined, its internal rate of return would be 82% in the expected scenario. Even in the worst-case scenario, its internal rate of return would be an acceptable 11%.

Words of Caution and Optimism

Given that we wish to provide results that can be generalized and a reliable framework to perform this type of cost-benefit analysis, let us look back at our analysis and identify the places it looks debatable, or overly conservative.

We believe that the most critical success factors that could not be assessed in this study have to do with scaling up the EPSS to the whole set of tasks covered in the training, and to the organizational structure and commitment to adjust the training accordingly. Although we are confident that the results would generalize to most of the procedural tasks of a CSR's job, the analysis included a training overhead cost that cannot be reclaimed unless a minimal proportion of the initial training is addressed and a re-engineering of the training department is conducted to adjust to the new situation. In other words, harvesting all the potential benefits requires a concerted effort among all parties—the software development department, the customer service department and the training department—to address the whole issue and make the appropriate adjustments.

Another critical factor that could not be precisely assessed in the current study is the maintenance cost. This factor was combined with the cost of capital and it was set at 10% of all costs, which is probably on the low side.

On the positive side, it must be remembered that this study did not include any benefits from increased employee performance, regardless of the initial training received. Surely a richer training environment that nurtures learning as exemplified by Hannafin's principles [5] should result in better performance over time. Such an environment could also have an impact on quality and even on employee satisfaction. We also mentioned earlier that the process of building an enriched training environment can help a company document and build its "knowledge capital." However, these results are harder to measure and require an expensive experiment spread across a longer time period. Nevertheless, the performance increase from this kind of continuous training can be very significant and outweigh other potential biases on the opposite side. Leaving this impact out of this study is a strong argument in favor of interpreting the results as conservative.

Another factor that makes this study conservative is the fact that EPSS technologies and tools are still at an early stage of development, and that time is on their side. We see that more and more tools to support learning are emerging in popular operating systems, such as Microsoft Windows wizard tools, bubble help, extensive online documentation tools, and Apple Guide, to name just a few of the existing tools now available to developers. These tools will contribute to decreased development costs and will increase the performance of EPSS tools that can be embedded in an application's interface. In fact, the software development team estimated that if the expensive hierarchical menu tool could be implemented with the WinHelp facility, almost no development cost would be incurred.

Recommendations from the Experiment

From a purely financial perspective, an expected break-even point of 4 years and an internal rate of return of 9% over five years would be considered a questionable investment for a risky project. And the wide gap between the optimistic and pessimistic scenarios leaves no doubt that it is a risky project. For such projects, the company would recommend a two-year break-even point. Or, taking another comparison perspective, capital risk investment firms look for projects with internal rates of return of 20% or more over no more than five years.

However, our study looked solely at the payoffs from the training perspective, which is only a portion of the benefits, the other sources stemming from increased productivity, better quality, knowledge capitalization, and worker empowerment. Moreover, there is a strategic decision involved: the potential of this technology is great and the company could wish to gain experience now instead of waiting for later.

Considering the other sources of payoffs and the potential of gaining experience through this project, we recommended going ahead with the project and implementing a process of data gathering and continuous monitoring of costs and benefits. This monitoring would help reduce the uncertainty of other projects by providing more empirical evidence on the positive and negative effects of the different factors involved. The continuous monitoring is also meant to provide indicators of when the costs can outweigh the benefits.

Another important recommendation was to include a member from the training department in the user interface design team. This involvement is crucial to re-engineering the training program so as to decide what material should be covered in the initial training sessions and what material could be left to the EPSS for the user to gradually learn on the job. Moreover, it was recommended that the EPSS be an integral part of the initial training for two reasons: It can improve initial training efficiency[5] and the user must get acquainted with the EPSS in order to use it effectively on the job. Otherwise the very first experiences with the EPSS may prove too tedious when a customer is waiting on the line, and as a result users might avoid the EPSS.

We also suggested that novice users are easier to assist than expert users and thus should be the first targeted users of the EPSS. Finally, the software development team suggested the use of standard tools such as WinHelp whenever possible in order to implement certain EPSS features. The use of such tools can significantly reduce the development cost, as was clearly demonstrated with the example of the "hierarchical menu" that was by far the most expensive technique to develop, but that could be reduced to almost no additional development cost if it could be implemented with the WinHelp facility. Of course, this is true only if the efficiency of both techniques is the same, which in our case was neither confirmed nor unconfirmed.

MICHEL C. DESMARAIS (desmarais@crim.ca) is the lead researcher at the Computer Research Institute of Montreal (CRIM) in Canada.
RICHARD LECLAIR (rleclair@pop.hydro.qc.ca) is a training specialist at a Canadian utility company.
JEAN-YVES FISET (jfiset@crim.ca) is a Ph.D. candidate at McGill College in Montreal, and the founder of Systeme Humain-Machine, Inc., Canada.
HICHEM TALBI (htalbi@crim.ca) is a research associate at the Computer Research Institute of Montreal, Canada.

Conclusion

Despite a number of success stories, EPSS remain a relatively new concept and little is known about their critical success factors. This study helps clarify what these critical factors are and provides new data on their assessment. It also provides a framework to analyze these factors and make predictions on their impact and on the outcome of an EPSS project. A striking result of the current study is that there was a factor of 17 between the break-even points of the optimistic and pessimistic scenarios. This represents a large span and leaves a lot of uncertainty in the project's expected results. It is probably due to the fact that many factors were modulated by the best/worst-case scenarios (software and pedagogical content development, training time reduction, maintenance) and that the co-occurrence of their worst- and best-case scenarios is very unlikely. Nevertheless, given that other factors were not accounted for (see the section "Words of Caution and Optimism"), this level of uncertainty does not appear unrealistic.

This level of residual uncertainty could leave a large number of profitable EPSS projects on the shelf if a cautious management is not willing to invest in new technologies. This is particularly true in light of the recent results on weak, and even negative productivity gains brought by information technology, which makes management more cautious of risky technology development projects [6]. This is a strong argument in favor of a more thorough study of cost-benefit analysis that can bring more evidence and empirical data on EPSS success factors. **C**

[5]As mentioned previously, EPSS have many advantages compared to traditional training. They adapt to the individual learner's pace, the learning context exactly matches the operational context, and represents an engaging learning mode in which the user has to actively seek the information to solve problems.

ACKNOWLEDGMENTS
The authors are grateful to Daniel Engelberg, Cameron Haynes, and Frances de Verteuil for their helpful comments. We also thank Daniel Daly for his contribution to the project.

REFERENCES

1. Carr, C. PSS! Help when you need it." *Training and Development Magazine* (Dec. 1991), 31–38.
2. Clark, R.C. EPSS—Look before you leap: Some cautions about applications of electronic performance support systems. *Performance and Instruction 31*, 5 (May/June 1992), 22–25.
3. Gery, G. Attributes and behaviors of performance-centered systems. *Performance Improvement Quarterly 8*, 1 (1995), 47–93.
4. Gery, G. *Getting Started: Electronic Performance Support Systems*. Gery Associates, Tolland, Mass.
5. Hannafin, M.J. Empirically-based guidelines for the design of interactive multimedia. *Educational Technology Research and Development 41*, 3 63–85.
6. Landauer, T.K. *The Trouble with Computers: Usefulness, Usability, and Productivity*. MIT Press, Cambridge, Mass., 1995.
7. Raybould, B. Performance support engineering: An emerging development methodology for enabling organizational learning. *Performance Improvement Quarterly 8*, 1 (1995), 7–22.