MANAGING EXPERT SYSTEMS FOR NEW FEDERALISM

William H. Fossett, Ph.D.
Office of Underground Storage Tanks
Environmental Protection Agency
Washington, D.C.

This paper expresses the author's personal opinion not Agency policy.

31 May, 1990

ABSTRACT: Change in the locus of operating knowledge that accompanies "New Federalism" policies implies different design and knowledge-engineering approaches for expert system applications. An expert systems strategy to implement a New Federalism approach in a Congressionally-mandated regulatory program is discussed. Tactical needs created by competition with conventional knowledge-delivery mechanisms are analyzed. Three knowledge-engineering tactics especially suitable for developing expert systems in this application environment are described.

* * *

Under "New Federalism" the locus of knowledge to operate Congressionally-mandated programs shifts from Washington to reside in state capitals. This change in the locus of operating knowledge has major implications for how Federal agencies design, disseminate and build expert systems. In "New Federalism" policy is developed nationally and translated into real operation by local agencies. In the Environmental Protection Agency's Office of Underground Storage Tanks (OUST) an expert system implementation strategy, designs and knowledge-acquisition tactics especially suited to a "New Federalism" regulatory program are evolving.

Major chunks of the knowledge to operate state tank-regulation programs develop and reside locally, rather than in Federal headquarters. However, other important knowledge -- e.g., involving scientific principles -- transcends localities in origin and in application. "New Federalism" recognizes the primary responsibility for discovery, development and diffusion of generally-applicable knowledge resides with Federal agencies.

To implement "New Federalism" OUST has modeled its national program after commercial franchising. Franchising works by using the strengths and knowledge of both the franchisees -- who directly run the business day-to-day and the franchiser -- who delivers national advertising, policy and crucial support services. Our expert system applications are a support service for states.

Major components of our strategy include expert-system development service to build administrative expert systems based on "local knowledge" from state agencies for local users; and, Federal expert systems based on "expert" scientific, legal or policy knowledge that deliver such Federal knowledge to local users.

Rapid development of such expert systems is tactically essential in this strategy. I apply "total quality management" techniques adapted from Japanese heavy industry to reduce our system development time and improve its quality. Three knowledge-engineering techniques whereby we speed-up the expert system development process will be described here: on-the-spot prototyping, brainpicking workshops and semi-automation of prototype review.

Capturing and Delivering Knowledge For "New Federalism" Governance

To serve state and local franchisees under a "New Federalism" approach expert systems managers have to muster two different kinds of expert system work:

- Expert systems development service which very rapidly captures administrative/local knowledge in small to moderate expert systems for use in localized settings; and,
- Expert systems application products which capture "outside expert" knowledge for use in state and local settings.

Expert System Development Service

There are over 50 states and other entities that implement Congressionally-mandated Federal regulations in the field. Each of these agencies covers multiple substantive-knowledge domains within a framework of individual state laws and regulations which take precedence when they are equally or more stringent than Federal laws or regulations. Particularly in environmental and general human welfare domains (e.g., housing, law enforcement, worker safety) the impacts of regulation are often felt most strongly at the local level, often making such domains politically controversial.
Although there is a legitimate Federal interest in such domains, most working knowledge of state laws, procedures and conditions is found within states, rather than at the Federal level. Recognition that this is the case as we build expert system applications for states is essential to cultivate political comity necessary to avoid regulatory stalemates. This recognition is also necessary to prevent state-Federal or state-state disagreement from becoming a barrier to an expert system's use.

Our prospective "customers" for these expert systems are state and local agency officials whose work is increasing more than their resources. Public concerns about human health and safety currently compete with public reluctance to pay higher taxes. The application of expert systems can assist state and local agencies to better meet the challenge of a workload greater than the resources available to meet public expectations.

Expert systems development service uses Federal resources to provide state agencies with very rapidly-developed, relatively simple expert systems applications that enhance the state's implementation and operation of its own regulatory program. This service captures local knowledge for expert systems that will support local administration of government programs. The knowledge captured originates in the local setting, not from "outside" experts. Such expert systems deliver local knowledge amongst local users to assure consistency and quality of local decisions.

However, successful execution of this approach depends on ability to build practical expert systems applications quickly and reliably enough to capture this opportunity. In many concrete situations in government the elapsed time and absolute cost to build an expert system -- rather than a favorable but ephemeral downstream cost-benefit -- will determine whether a system is built. If the cost seems too high or, development time likely to be too long, government managers are likely to choose conventional tools -- print media, videotape, training workshops/tools -- rather than expert systems for knowledge delivery. The decision calculus involved is not that of comparative costs and benefits of expert systems versus convention knowledge-delivery media. Instead, such decisions involve other dynamics that influence management decisions in any complex organization.

In service applications development there are relatively narrow boundaries of elapsed time and cost to be honored lest this component of the strategy collapse under its own weight. For example four applications for each of 56 different states and trust territories would be 224 applications to develop. This poses a special challenge to expert system development contractors and bureaucrats alike to move fast and minimize errors, lest costs balloon above absolute thresholds.

Tight control of elapsed time and cost for development of practically-useful expert system applications is thus crucial to the success of this strategy. However, such control has been very difficult to achieve in the development of any kind of software, including expert systems applications. In pursuit of continuous improvement in the performance of the process whereby OUST produces expert system applications I identified several points where we could significantly reduce and improve control of development time: 1) initial prototyping; 2) knowledge acquisition; and 3) user review of prototypes.

An alternative to a build-it-yourself expert system diffusion strategy. This "development-service" approach to building expert system applications is an alternative to the "build-it-yourself" development of expert systems applications exemplified by Ed Mahler's now-legendary strategy at DuPont. DuPont has long-term, technically-expert employees with sufficient time and tools at hand to learn how to develop their own expert systems, to do so, and, to then support those systems. State and local government agencies typically do not.

State regulatory agency staffs are typically relatively small, have relatively heavy workloads with little spare time or equipment. State environmental agencies report high employee turnover rates as staff skills grow and become much more valuable on the private market.

A DuPont-like each-one-build-one expert system strategy is likely to fail where turnover is high and time is short because those grassroots experts lack the time, the software, the attention and the hardware-access to learn how to, to actually build, and, then, to implement an expert system application.

In state agencies there is a high premium on getting the organizational results one could wring from expert systems technology, i.e., getting the regulatory work done right the first time, in a timely manner, within resource constraints. In such settings the build-it-yourself approach to implementation of expert systems is in reality likely to fail for lack of sufficient in-house resources.

"Federal-Knowledge" Expert Systems Application Products

State agencies also seldom have sufficient resources to develop or evaluate basic scientific or technical knowledge needed to implement a program. This knowledge has traditionally been delivered to agency staff via Federal guidance documents, training courses and sometimes videotapes. In turn, the state agency employees is expected to absorb, retain and at the right moment apply this knowledge. At any point this flow of knowledge can easily break and harm the quality of local decisionmaking. The added work required to check or correct decisions made this way seriously harms government productivity. Expert systems can deliver "general" knowledge directly to the point of decisionmaking without risk of the breakdowns that occur when we rely on the conventional methods of knowledge delivery.
Expert system application products are Federal-ly-developed and managed expert systems that capture and deliver centrally-held or developed knowledge -- knowledge that crosses the bound-aries of state laws and local knowledge. A Federal expert system product captures and delivers knowledge -- especially, scientific, legal or policy knowledge -- more sophisticatedly and more con-ceived in expert systems literature. Such a Federal expert system delivers knowledge to local users that originates outside the local-use set-ting. For example, we took knowledge developed by expert EPA research laboratory about three soil tr-eatment techniques and repackaged it in an expert system application that enables an inexpert user to estimate the possible success of each technique at an individual corrective action site.

Recruiting Volunteer Expert System Users In A "New Federalism" Setting

A "New Federalism" approach allows the Federal agency much less power (and opportunity) to di-recct or coerce state agencies to do things in a particular way. State agency officials are re-luctant to invest scarce local staff time to par-ticipate in developing an unfamiliar computer technology -- such as expert systems. These "cus-tomers" modal experiences have been with data-base management or algorithmic modeling software that took too long, cost too much, was hard to use and often did less than they hoped. Many Federal agencies also have Federal regional offices placed between the expert system developer and the intended state-agency user. These conditions tend to protract the elapsed time required to build an expert system application, but are not reme-died initially by better system-development techniques.

User-involvement in design is not sufficient alone. When we set out to develop our first ex-pert system application we very closely involved a panel of prospective users from several states in its design and testing. Initially we assumed that the prospective users of our first expert system would see its value and voluntarily use it. Al-though the system functioned well, we discovered that the seemingly minor differences in approach from one state to another prevented this com-promise system from being specific enough for use by any of the individual states that had contributed to its development. Had we been implementing "old Federalism" we could have simply exerted our central power and required states to use the application. However, under "New Federalism" such coercion would poison the well of good will we were building with state agencies.

The most important goal at this juncture was to get someone to use the expert system instead of conventional knowledge-delivery tools, so that we could begin producing empirical evidence of its value. Thus we needed to recruit several states who would work directly with OUST to fine-tune the existing expert system to make it usable and visible in those states. I would identify some prospects and go demonstrate our prototype expert system application in hopes of "selling" them on using it in real work in their state.

"Customer" response to prototypes triggered this strategy.

The evolution of the two-pronged strategy de-scribed here was triggered by the candid re-sponse of a state-program director to a demon-stration of our system prototypes: "Those are real nice Bill, but they don't match the problems that worry me most about my program. I could really use software like this, but we would need it to do different work than these do."

He identified two topics of immediate concern to him and introduced me to his staff responsible for them. If my customer did not want what I had in my sample case, then I would make what he did want. Though the topics involved were moderately complex I volunteered to prototype them on the spot.

One person had already laid out one topic as a decision tree which I copied and took back to the hotel. In several hours I had a working prototype that captured this decision tree. The following day I secluded myself in the hotel with my laptop and hacked away at a prototype of the other topic. And though it was more complex, and the underlying knowledge much less stable, I estimated that this too could be captured and worked into a system in several weeks working time.

A day later I returned and demonstrated my prototype to the program director and other staff. Their response was very positive; they felt the software would be useful in implementing this part of their regulatory program. They would agree to distribute it for use in their 16 field offices if it would be made to capture the final version of the decision tree which would be approved by the state commission overseeing this policy. I agreed to this; and, we also agreed to take the steps needed to formalize a partnership between the state and OUST for development and showcasing of other expert systems to other states.

This experience taught me that we could build "custom" systems of a practical-value complexity and magnitude much faster and easier than we could build systems tailored for individual states.

It would take much more time and effort just to persuade different states to agree to do a task the same way in order to use the same expert system than to build individual systems tailored for indi-vidual states.

Relatively small, custom-tailored expert systems would have to provide the user with obvious, immediate added value for their time and effort in using the software to successfully compete with other demands on users' attention, as well as with other knowledge-delivery mechanisms. If scoped too narrowly, even a custom-tailored system
would not attract users readily.

However, I believe neither local-knowledge nor "universal-Knowledge" applications alone would attract enough system-use in the daily work-environment to be effective administratively. And, we are serious about bringing Federally-produced scientific and technical knowledge to bear on locally-made decisions. To accomplish this special task, the expert system application so that it weaves itself into the ordinary patterns of working of its intended users. Each use of the expert system must add value to the work that the user contributes -- not just be another hoop to jump through.

Expert system application design in competition with conventional knowledge-delivery media.

Meanwhile other potential expert-system users were asking for (and getting) conventional knowledge-delivery tools to do the work of the sort that our first expert system originally addressed. The knowledge vacuum which expert systems applications were intended to fill was in fact being filled with guidance documents, technical papers, videotapes, etc.

Though generally very well-done, our conventional tools have a serious inherent weakness of all their kind: they rely on highly variable human attention, understanding and memory to absorb and recall the knowledge they purvey. Unlike expert-system knowledge, the expert system videotapes and training go partially-read or partially-understood, and when the time for application comes, partially-remembered.

This endemic failing of conventional knowledge delivery mechanisms does not figure into decisions to develop them however. If any expert system were as unreliable a mechanism for knowledge delivery as the best hardcopy publication, videotape or training workshop, that system would not even be considered an acceptable prototype. Although the hurdle for expert systems is higher, that higher standard will serve the public better in the long-term if we can clear our higher hurdle.

Meanwhile, expert system developers must over-achieve to compete effectively with conventional knowledge-delivery mechanisms to deliver the knowledge would-be customers are shipping out in the media familiar to them.

In a New Federalism setting the practical truth is that your expert system applications must be perceived as patently easier and more rewarding to develop and use than the conventional tools of knowledge delivery. The advantages expert systems offer over conventional knowledge delivery tools are not yet known and believed first-hand by most prospective users and sponsors.

Will an expert system cost more or less, be finished sooner or later, than a guidance document or a videotape or a series of training courses to convey the same knowledge? Though much emphasis has been given to projecting the costs and benefits of building an expert system application, you will find that this may not be the most important practical comparison applied in decisions to use an expert system to deliver knowledge.

Successful competition with other knowledge-delivery media requires that expert systems applications be designed:

- To be conspicuously easier to use than a book;
- To provide obvious, added-value immediately to the user;
- To be at least as accessible as books or videotapes; and,
- To encourage end-user contribution of knowledge to improve the system and keep it fresh for continued use.

These design characteristics must be reached within time and spending limits that are equal to or better than those for competing knowledge-delivery media. This applies to local-knowledge systems and "outside-expertise" based systems alike.

It is not sufficient merely to describe these design features and promise their advantages. You must also be able to deliver them in real applications faster and cheaper than an agency can field a series of training workshops or videotapes. Speedy development and demonstration of a working prototype can be persuasive. But if subsequent development of the system lags (or costs grow too high) then the client organization is very likely to revert to the familiar knowledge-delivery tools.

Organizational -- rather than technical -- sources of delay can significantly increase development time and cost to build an expert system application. Thus it is crucial that knowledge engineering techniques applied in this situation minimize occasions for organizational delay -- e.g., in user review of prototypes.

Rapid Knowledge-Capture Techniques Developed in OUST

A rapidly-developed proof-of-concept prototype can help you to estimate the cost and complexity of the full system and to develop a persuasive, credible presentation for it. However, if this proof-of-concept prototype is relatively dumb -- i.e., delivers knowledge that its prospective sponsors believe to be trivial or obvious -- your prospective sponsor or user will quite properly say, "We can tell them that in a simple pamphlet if that's all this software is going to do." Or, "If that's all it does it's easier to just look the information up in the guidance rather than go over and use the computer." Underdeveloping the proof-of-concept prototype can cost you the opportunity to build the system. Making a proof-of-concept system smart enough to get a sponsor or user to...
Much of environmental regulation involves application of specialized scientific and technical knowledge that is constantly growing. A healthy level of disagreement among experts is part by-product and part cause of this growth in our knowledge. However, expected or apparent disagreement among multiple sources of knowledge in an expert system can delay production of an expert system. The "brainpicking workshop" can enable you to work through such disagreement quickly and cleanly.

In a brainpicking workshop a panel of "experts" builds a set of examples from which a decision rule is induced using an expert system shell. Any shell which offers an inductive algorithm -- typically ID3 -- such as lstClass FUSION, VP Expert, KnowledgePro or others, may be used in this manner. My personal preference is for lstClass FUSION because it displays the induced rule as a decision tree which the panel can quickly observe and discuss.

To conduct a brainpicking workshop you need a comfortable, isolated setting, a very large display screen so that everyone is looking at the same display as a group, and a PC with a printer. You should prepare preliminary "decision matrices" for each domain you expect to address in the workshop which show the factors to be taken into consideration and the decision (or advice) alternatives. First you will ask the panel to confirm or correct your draft "decision matrices," taking only one domain at a time.

After confirmation/correction of each domain's "decision matrix" is completed you then lead your panel in a process of filling in the "decision matrix" with examples -- combinations of factor-values which may be used to arrive at an advice or a decision alternative. You start out with the clear-cut examples and as the panel becomes accustomed to working together you move into the more ambiguous cases. All examples are entered into the system live, on-the-spot, not later on. As the set of examples grows you periodically (for example, during breaks) induce and display the decision rule for the panel to review.

The use of structured examples opens the way for identifying and resolving disagreements among experts at one time and place to supply the basic knowledge for an expert system application. It reduces system development delays due to communications lapses with knowledge sources and, to perceived disagreements between experts' judgments.

Expert system developers' tools and techniques carry an implicit assumption that there is "the expert" whose domain knowledge and authority are definitive. This assumed condition is seldom found in complex organizations, especially government agencies, where knowledge and authority are distributed across multiple individuals in different places with different schedules that may pose daunting knowledge-capture delays. The brainpicking workshop can also help you to capture distributed expertise more quickly.

buy-in depends on your ability to rapidly capture at least moderate depth of knowledge in the proof-of-concept system.

On-the-spot rapid prototyping. Being amidst the work of the prospective sponsors and users of an expert system application is an invaluable opportunity to capture knowledge. A proof-of-concept prototype can build "in the field" -- i.e., on a laptop, in a hotel room -- in 8 to 12 hours with very good effect. By staying nearby the local knowledge sources, the developer is positioned to quickly capture additional knowledge necessary to resolve basic ambiguities which might otherwise make your prototype look trivial or dumb. Within two days you have the proof-of-concept prototype back in front of your prospective user and/or sponsor for a go/no-go decision regarding development of a testable prototype.

If you do this from afar, your time and costs at this stage are likely to increase significantly. The knowledge engineer will spend time trying to infer logic that could be had for the asking if he were face-to-face with the knowledge source. And, being physically present for a known-limited period of time enhances your chance of success in reaching a knowledge source. (Finally, if you are sufficiently tactful about this, quick, precise consultations with knowledge sources can also enhance the receptiveness of potential sponsors or users for the prospective system.)

This pattern of working, however, imposes technological constraints on your development tools. Whatever development software you are using must work in a laptop. A shell which minimizes the opportunity for developer error is essential, especially one that has some sort of graphic knowledge representation which you can modify directly minimizes the opportunity for syntax errors. And, if appropriate (not necessarily) a graphic knowledge representation can be displayed much more persuasively to substantive experts or state-agency managers than can lines and lines of syntax-intensive code.

The brainpicking workshop. The brainpicking workshop physically convenes a panel of recognized experts at one time and place to supply the basic knowledge for an expert system application. It reduces system development delays due to communications lapses with knowledge sources and, to perceived disagreements between experts' judgments.

Expert system developers' tools and techniques carry an implicit assumption that there is "the expert" whose domain knowledge and authority are definitive. This assumed condition is seldom found in complex organizations, especially government agencies, where knowledge and authority are distributed across multiple individuals in different places with different schedules that may pose daunting knowledge-capture delays. The brainpicking workshop can also help you to capture distributed expertise more quickly.
tunity for human judgment in ambiguous or sen-
sitive cases.

Although it is possible to draw (or type) decision rules directly into the system, I believe that to do so limits ability of this tactic to work through disagreement. Keeping your panel focused on concrete cases rather than on personal beliefs or theories is a much more reliable path towards agreement. However, when you are working with one "expert" or knowledge source the direct construction of decision trees can be very effective in rapid capture of knowledge.

Semi-automated review functions for deepening or reviewing prototypes. User or expert review of your expert system prototype is another point at which elapsed time can get out of control and significantly slow development of the system. Establishing level of agreement with the advice or decision an expert system recommends is a fundamental element of its review. Level of agreement can be captured semi-automatically by designing "agreement windows" and pop-up comment windows into your review prototypes.

When the review prototype offers advice, a dialogue window pops up that asks the user whether or not he agrees with the advice/decision that has just been recommended. If the user selects "disagree" a second screen pops up which lists the user's answers and the advice they led to. Below this there is a menu of alternative "advice" the system could have given instead. Instructions on-screen direct the user to select the advice which he believes would be appropriate under that set of conditions. On completion of this window, the user sees another dialogue box offering to open up a note window for any lengthier comment the user may wish to make. An option to pop-up a note window is also included on every data-input screen in the review prototype.

Data from user responses to these dialogue boxes are written to simple database files that can be analyzed with a spreadsheet or other analytic tool to determine level of agreement. In cases of disagreement you can feed the user's case back in to the inductive system to identify examples with which it conflicts or to test the effect of including this particular case on the rule. Written comment files are stored as simple text files named according to the particular screen or factor they refer to. Soliciting the reviewer's input this way structures that input and does some of the work of preparing it for your analysis. It aids in focusing your reviewers' attention. If you want a broader look at the system, that can be gleaned through the more traditional style of consultative oral interview.

If you want to forestall procrastination among reviewers you may consider building a time-limit function into the prototype that disables the prototype on a certain date. As the end of the review period draws near, this function opens a window on screen which indicates the number of days left for review. Finally when the review period is ended, this function deletes one or two key files necessary for your prototype to operate, thereby disabling it.

The expert system functions necessary for you to do this are available in many expert system shells such as lstClass FUSION and KnowledgePro, or other systems that offer hypertext capability. No doubt they could be implemented in more elaborate development tools also.

Expert Systems and New Federalism Policy

Habits of thought -- such as old-style Federalism -- tend to permeate and shape the way we design and develop information systems. Those systems in turn exert a pervasive influence on Federal and state governmental relationships and responsibilities which New Federalism aims to improve. When we recognized that the conceptualization of our expert system strategy was inconsistent with our relationship with the state agencies on which our success depends we changed our expert system strategy. This strategic change entailed development of the knowledge-engineering tactics which have been described here. There is much more to learn, no doubt, as our adventure in New Federalism unfolds and many more opportunities to reduce development time and improve the quality of our work. Though it is far too early to claim success for our expert system strategy I am confident that it is now headed in the right direction.