SLURRYMINDER: A Rational Oil Well Completion Design Module

E. Brent Kelly, Philippe Caillot, Robert Roemer, and Thierry Simien, Dowell Schlumberger

A critical phase of oil well completion involves positioning cement between the outer surface of a metal casing and the sides of the well. This task is done by injecting a specially formulated cement slurry down the center of the casing and up the sides of the bore hole. Designing these slurry systems is time consuming and expensive because of the variability of the conditions between wells and the variability of the raw materials and techniques used in geographically diverse locations. SLURRYMINDER is a design tool to aid field engineers in creating globally consistent cement slurry formulations and to rapidly disseminate current well-cementing techniques. We describe the implementation of this system and why AI technology was used; we also discuss corporate benefits of the system, both real and projected. We provide details on the SLURRYMINDER development process, its worldwide deployment, and our experiences in maintaining and updating it.

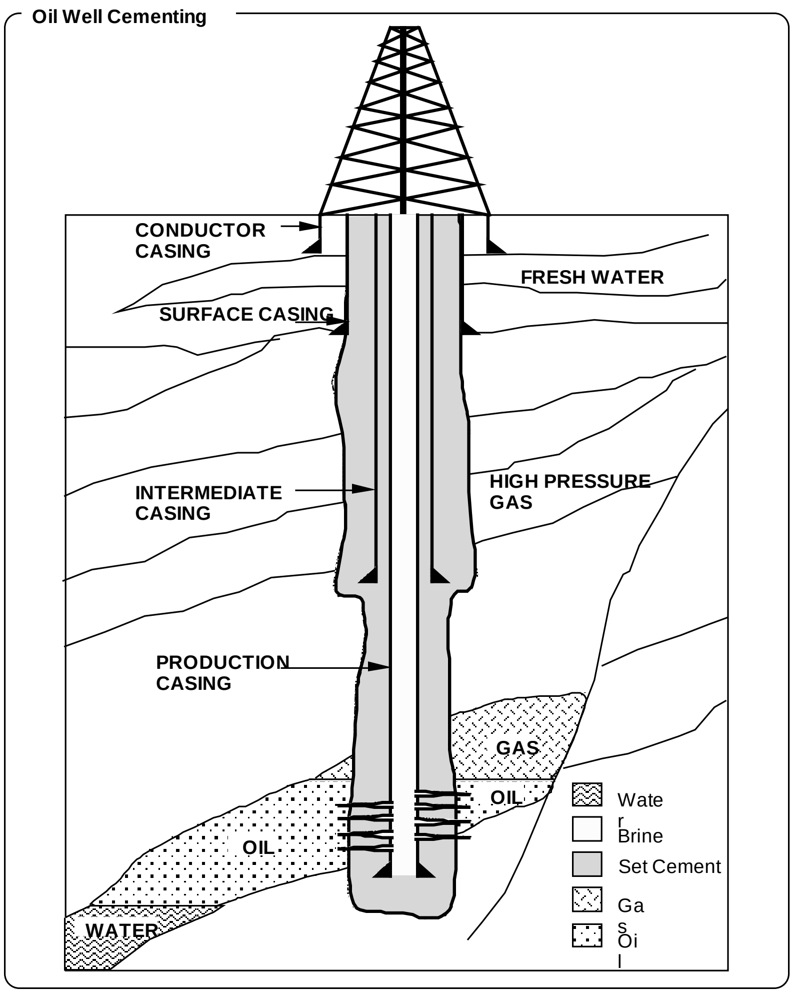


Figure 1. A Schematic Showing How Oil and Gas Wells Are Cemented. Cement slurry is pumped down the inside of the metal casing and up the annulus formed by the outer surface of the casing and the sides of the well bore hole.

Problem Description

A critical phase of oil or gas well completion involves positioning cement within the well annulus by pumping a specially designed cement slurry down the well casing and up the sides of the bore hole, as shown in figure 1 and discussed by Nelson (1990). A typical cement slurry is composed of the cement powder, some type of mix water (usually fresh water or seawater), and one or more chemical additives that give the cement certain specified physical properties during pumping and afterward when set.

Designing a cement slurry system is generally iterative in nature: An initial design is proposed, which is then followed by a series of laboratory testing and tuning steps, as in figure 2. During this testing-tuning process, additive concentrations can be modified or certain additives replaced by others until the slurry and the set cement satisfy the required physical properties, whereupon it is ready to be pumped into a client's well.

Ideally, given similar well conditions anywhere in the world, slurry designs should contain similar additives at similar concentration levels. In practice, because of the variability in the cement powder, the quality of the local additives, and local cementing tradition, slurries designed for similar well conditions often bear little resemblance to one another. This dissimilarity not only can be a source of confusion to clients, but from a global research and engineering perspective, it also makes it difficult to disseminate new cementing additive technology.

To address the slurry design problem, Dowell Schlumberger's engineering personnel embarked on an ambitious program to create a slurry design support tool. Our goals in developing this tool were twofold: (1) to ensure worldwide design consistency and quality but allow the required freedom for local practitioners and (2) to create a distribution mechanism for rapid information dissemination of an ever-evolving technology. Earlier work in designing entire cement jobs had already been done by practitioners, such as Wolsfelt, Roger, and Fenoul (1989); however, in their CEMENTEX system, no attempt was made to design the critical cement slurry systems that are actually pumped into the well.

Prior to investigating a solution to the slurry design problem using AI techniques, earlier attempts to solve the problem were made using statistical regression analysis and material characterization studies. Researchers tried to analyze existing field-design data to develop correlations between slurry performance properties and the amount of each chemical additive in the slurry mixture. Unfortunately, because of the variability in the local cement and chemical additives, the deviation in the regression parameters was so large that the parameters had no real significance, and these efforts failed. Our field engineers and laboratory technicians were, however, successfully designing cement slurry formulations daily, and we became convinced that an AI approach was the only feasible alternative remaining if we wanted to achieve a solution to this problem.

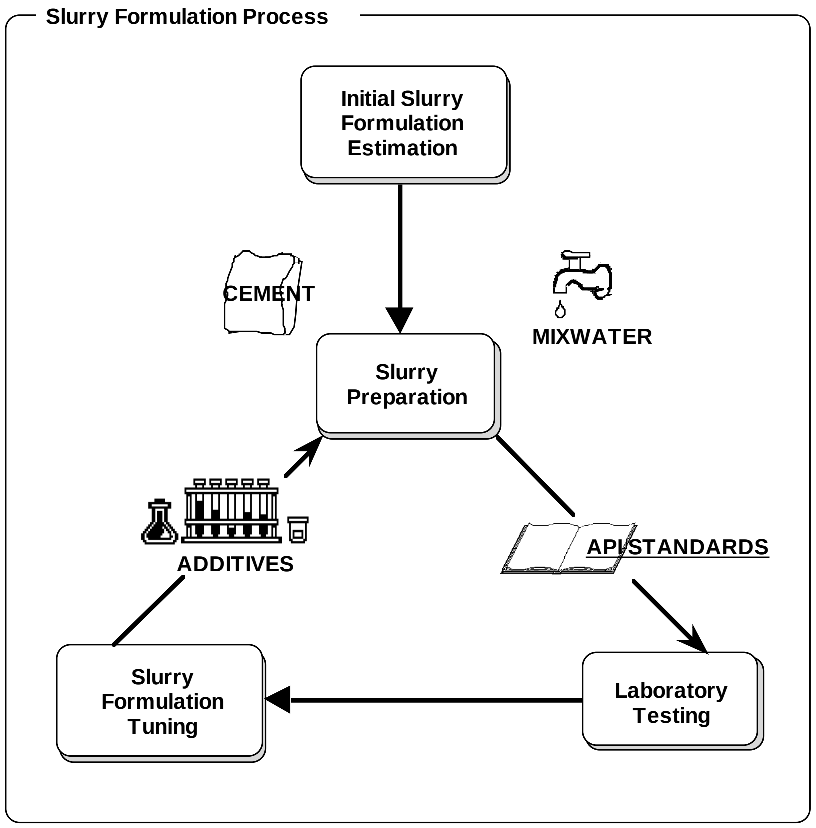

Slurry Formulation Process
Initial Slurry Formulation Estimation
CEMENT
MIXWATER
Slurry Preparation
ADDITIVES
API STANDARDS
Slurry Formulation Tuning
Laboratory Testing

Figure 2. The Cement Slurry Formulation Process.
An inital design is formulated and given to the laboratory personnel. This design is mixed in the laboratory, and a series of standard tests defined by the American Petroleum Institute are run on the resulting slurry. If the required slurry physical properties are not achieved, the slurry formulation is tuned by modifying the concentration of one or more of the chemical additives. This tuned slurry formulation is then mixed, and the tests are rerun. This testing and tuning loop continues until the slurry satisfies the performance parameters, as specified by the client or the sales engineer.

Application Description

By nature, successful cement slurry design is a fuzzy domain where heuristic information and experience about cement and additive use is required. Compounding the problem is the fact that some of the heuristics change depending on geographic location: Design experi-

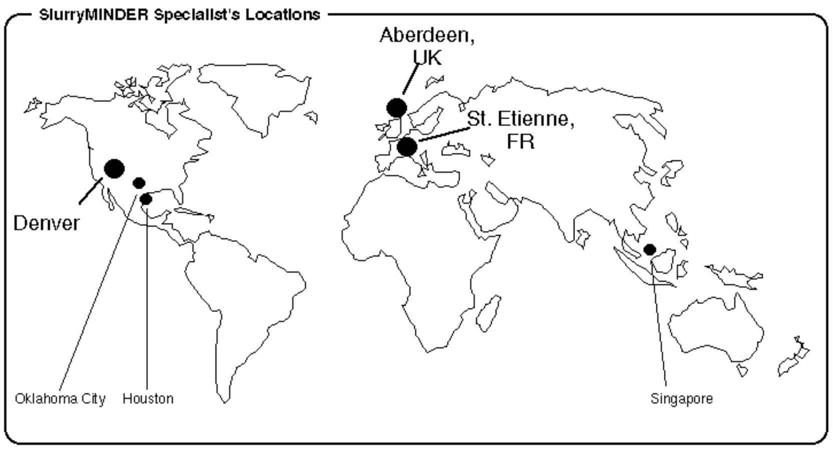


Figure 3. Cementing Specialists Who Assisted with SLURRYMINDER Development Were Located All over the World.
Primary specialists involved throughout the development process were located in Denver, Colorado; Aberdeen, Scotland; and St. Etienne, France. Additional experts providing input were stationed in Singapore; Oklahoma City, Oklahoma; and Houston, Texas.

ence in one field location might not be valid in another. As these problem characteristics became known to the development team, we concluded that solving the slurry design problem would likely require the use and integration of an extensive amount of information from a variety of sources. Having had significant experience within Schlumberger using AI techniques, we knew many of the risks and benefits associated with these technologies and opted to approach the solution to our problem through the creation of an imbedded knowledge-based system.

Development Constraints

While designing and implementing SLURRYMINDER, we were required to work under several existing constraints.

Distributed expertise and varying methodology: Our cement slurry design specialists were distributed all over the world, as illustrated in figure 3. Although they were all experts in their own region, their design methodology and design experiences were somewhat different. To rationalize these various points of view, one specialist at the engineering center in St. Etienne, France, was appointed knowledge czar, with the responsibility of converging methodologies and making strategic decisions when conflicting opinions arose.

Off-the-shelf software tools: Funding and timing constraints would not allow us the luxury of creating a custom expert system shell or developing a custom human interface tool. Hence, we were required to use software tools that were available in the software marketplace.

Existing design data: At all company locations, a cement slurry design database exists as a repository for successful slurry formulation data. We wanted to somehow incorporate the use of this local database into SLURRYMINDER so that the global methodology could be made to apply in each locality.

Computer hardware specification: In the mid-1980s, each of our 155 field locations purchased a MICROVAX II with a VT-240/241 semigraphic display. To effectively use this investment, SLURRYMINDER was required to execute on the existing field computer hardware. The implication of this decision was that the interface had to be character based.

Shell Selection Criteria

Prior to selecting a shell for use in constructing SLURRYMINDER, we performed a domain analysis to determine which kinds of knowledge were to be represented. Using this domain information and making some pragmatic decisions, we formulated the following selection criteria that we used in the evaluation and selection process:

First, it must support knowledge representation paradigms, including a robust inference engine that allows both forward- and backward-chaining production rules and an object-oriented schema system that supports multiple inheritance. The schema system must be well integrated with the inference engine pattern matcher so that the rules can effectively use the objects, slots, and slot values in their antecedent or consequent parts.

Second, it should support the concept of multiple knowledge bases, allowing knowledge decoupling and grouping.

Third, it must support the integration of user-defined procedures, functions, or routines in both the left- and right-hand sides of production rules (which implies calling-out functionality).

Fourth, it must be able to be embedded within a larger software system or program (which implies calling-in functionality).

Fifth, it must provide a development environment with the following minimum capabilities: dynamic rule and object creation through rule and object editors; browsing features to examine rules and objects within the knowledge base, including progeny graphs, rule-network graphs, and textual listings of rules and objects; breakpoints to control rule execution; and tracing features to observe rule behavior and knowledge base modification during execution.

Sixth, it must provide a small run-time kernel that can be used to execute systems created with the shell without the development environment present. In this run-time environment, system performance should improve.

Seventh, it must support a run-time kernel that does not require a bit-mapped graphics terminal or additional software packages to execute properly. The human interface should be determined by the application developers and not the selected shell.

Eighth, it must execute on a variety of hardware platforms, with knowledge bases requiring only minor modifications when changing platforms.

Ninth, it must have a significant user community to ensure its robustness and reliability.

Tenth, it must be able to operate in conjunction with future acquisition and design software written in C.

Eleventh, it must be accompanied by clearly written and professionally typeset documentation. This documentation should be written so that individuals reasonably familiar with software development can understand how to use the shell without having to ask for undocumented functions and features.

Twelfth, it must be priced so that development copies can readily be purchased by the company's engineering centers. The run-time version must be priced so that a worldwide license can be obtained economically.

Thirteenth, it must be able to be called by an Ada program or be linked with Ada program modules.

Fourteenth, it must allow communication with an INGRES database through query facilities within the shell or through the calling-out–calling-in functions.

In addition, the company or organization marketing the selected shell must do the following:

First, it must provide a hotline service staffed by qualified professionals available to answer questions regarding use of the shell. A fee can be charged for this service.

Second, it must be able to provide training either in house or at a remote location to teach individuals how to use the tool. A fee can be charged for this service.

Third, it must be reputable, with at least five years of experience in developing expert system shells or other AI software.

During the selection process, we evaluated three C-based expert system shells; our final choice was NEXPERT OBJECT from Neuron Data.

System Design

A simple representation of the SLURRYMINDER architecture is shown in figure 4. The application consists of five distinct parts: (1) human interface routines; (2) utilities for system configuration, browsing of other useful online databases, and a slurry pricing spreadsheet; (3) the reasoning mechanism for generating the list of chemical additives required in the slurry formulation; (4) the explanation and warning subsystem used to explain why a particular formulation was generated and whether there are any warnings on how to use one or more of the additives in the formulation; and (5) the query and calculation routines for generating additive concentrations and the quantities required for laboratory testing.

Knowledge Base Decomposition and Design. In attempting to create a model of how our specialists approach the slurry design problem, we became aware of four distinct levels of abstraction in their thinking processes: (1) recognition of which type of slurry system must be generated to satisfy the required performance parameters, (2) recognition based on the chemical product family that can be used in the slurry system, (3) an evaluation of whether a particular chemical family is actually required in the formulation, and (4) the actual selection of one or more of the chemical additives from a particular family that enables the slurry to meet one or more of the specified performance parameters.

Knowledge within SLURRYMINDER was decomposed into 14 separate knowledge bases, 1 knowledge base for general or kernel knowledge and the others for specific knowledge relative to one particular chemical additive family. Abstraction levels one through three are contained in the kernel knowledge base, but level four is contained within the knowledge base for each particular family, as in figure 5. Representing the current level of knowledge requires approximately 115 classes and objects and over 700 rules, with several C routines used for optimizing dynamic object creation during inferencing.

When a user queries SLURRYMINDER for a slurry formulation, the reasoning mechanism uses a backward-chaining, depth-first search strategy. We implemented an exhaustive searching strategy so that multiple types of designs can be obtained if applicable, with each type of design potentially having multiple formulations. During inferencing, competing slurry formulations are ranked according to a simple model based on how well a particular additive will perform in a given situation. Users can limit the number of solutions that are generated by SLURRYMINDER, which in certain instances can grow geometrically with the number of selectable additives at each formulation stage.

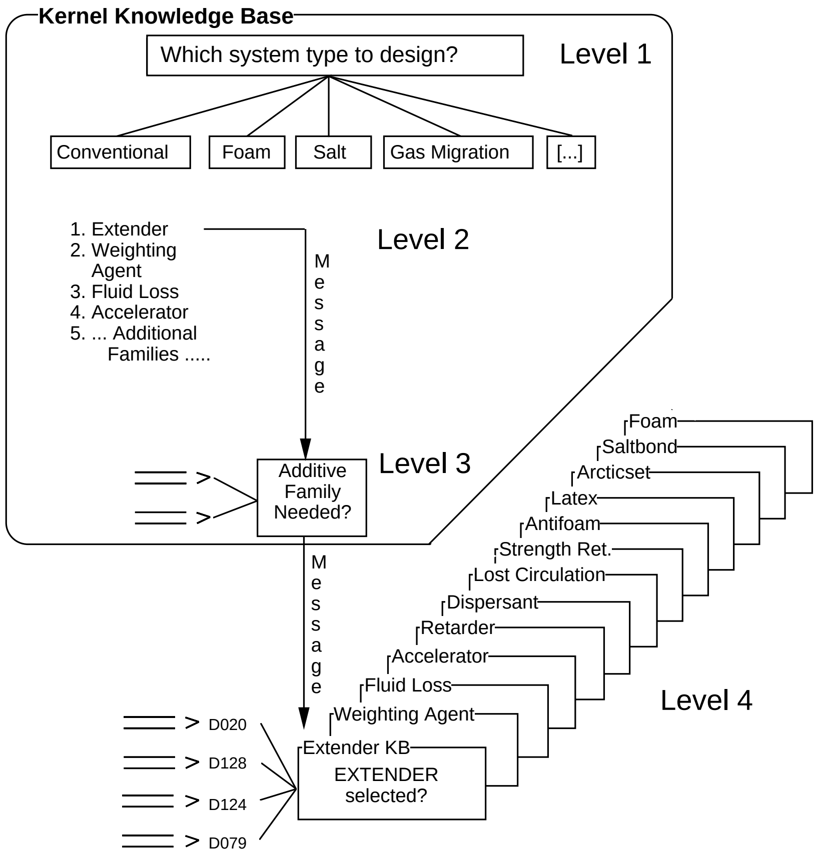


Figure 4. Simple Schematic of the SLURRYMINDER Architecture.
Five separate subsystems are connected by a set of system kernel routines: the human interface, the inferencing mechanism, the concentration routines, the explanation and warning subsystem, and some utility routines for database browsing and administration. Within the inferencing mechanism, there are 14 knowledge bases containing over 700 separate rules and solution strategies.

We designed the knowledge base so that each chemical additive family is self-validating about its applicability in a given design problem. When a family determines that its functions are required in slurry formulation, it sends a selection message to each individual chemical additive within the family. Individual additives are also self-validating; when they receive the selection message from their parent object, rules are evaluated to determine if the particular additive can be selected given the current state of the formulation and the global design pa-

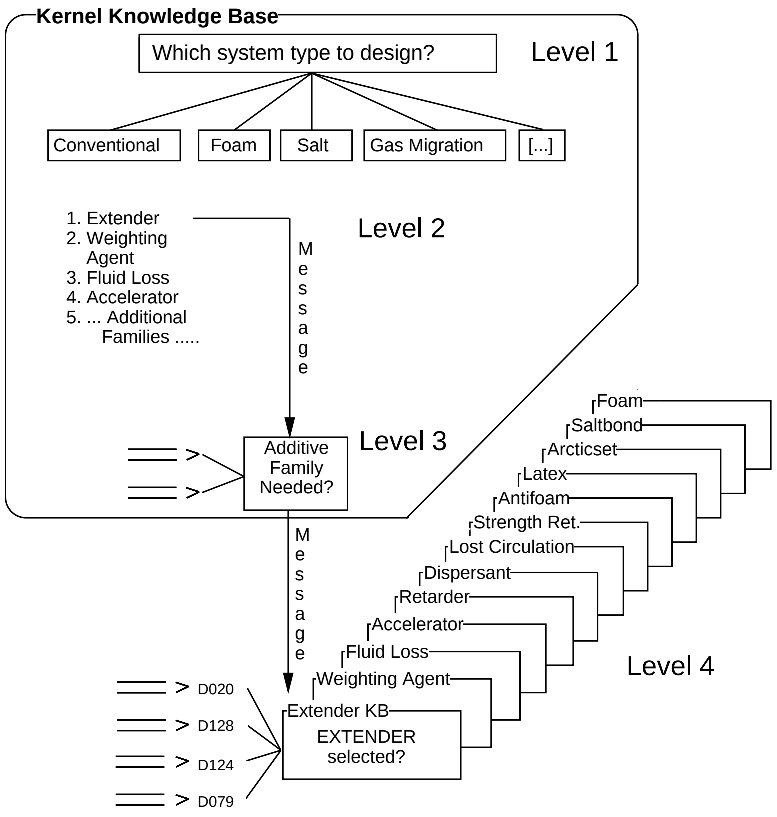


Figure 5. Knowledge-Abstraction Levels within SLURRYMINDER.
Level 1 determines which type of system to design. Level 2 is a declarative level, indicating which additive families might be included in a particular system type. Level 3 is activated by a message from level 2. Level 3 determines if the effect of a particular additive family is required given the input data and the additives already selected. If the effect of an additive family is needed, level 3 sends a message to level 4, which contains the selection knowledge for each particular additive in the family. Levels 1, 2, and 3 are contained within the kernel knowledge base, and level 4 is divided into 13 separate knowledge bases, one for each additive family.

rameters, such as well temperature or depth.

As additives are selected for use, they append an explanation code and a warning code to the current intermediate solution node, as illustrated in figure 6. These codes can then be used by the explanation and warning subsystem to extract context-specific explanation or warning messages from a flat-file database specific to the additive in question; thus, users are informed about why a particular additive was selected in a given situation. It is this explanation and warning system that provides the mechanism for rapid technology transfer within SLURRYMINDER. When new chemical products are added to the SLURRYMINDER knowledge base, they appear in the ranked formulations they obtain from the reasoning mechanism. After a particular slurry formulation is generated, the user can invoke the explanation and warning option to view an explanation of how this new additive is to be used. If more information is desired, users can branch using a hot key to a more complete online Additive Information Manual, giving complete technical details for each chemical product.

Previous Design Database Link. Users of the SLURRYMINDER prototype informed the development team that slurry formulations without additive concentrations were useless. Linking with the local design database is where SLURRYMINDER bridges the gap between a global formulation methodology and local practice.

Once the additives in a slurry formulation are selected, users have two options for generating additive concentrations: a default method or a query of their own local database, as shown in figure 7. The default method can be likened to browsing a cementing manual for general recommendations for additive concentrations, independent of the local cement and additive quality. Querying a user's local database, however, provides a connection with previous design history in the same geographic location.

When users query their local design database, interactions between the local cement and the local chemical additives are implicitly accounted for because these previous designs have already been tested successfully in the laboratory. Queries usually consist of some design input data, such as temperature and density; the targeted physical properties of the cement; and additional items for narrowing the search, such as the well name or the client name. These criteria are used to build an SQL-like query to search a series of relational tables in the database. The output of the query consists of the average chemical additive concentration value for all tests matching the query criteria, the minimum and maximum values, and the total number of tests found. Users can review this information and reformulate the query if narrower criteria are needed.

Linking with previous design experience through these local databases leverages and validates the corporate strategy for a global design methodology that can be made to apply nearly anywhere in the world. It is this coupling of expert system technology for selecting the chemical

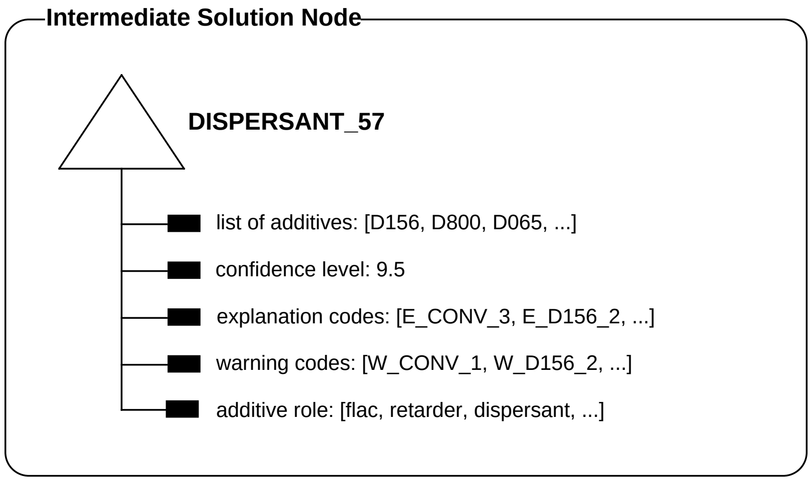

Intermediate Solution Node
DISPERSANT_57
list of additives: [D156, D800, D065, ...]
confidence level: 9.5
explanation codes: [E_CONV_3, E_D156_2, ...]
warning codes: [W_CONV_1, W_D156_2, ...]
additive role: [flac, retarder, dispersant, ...]

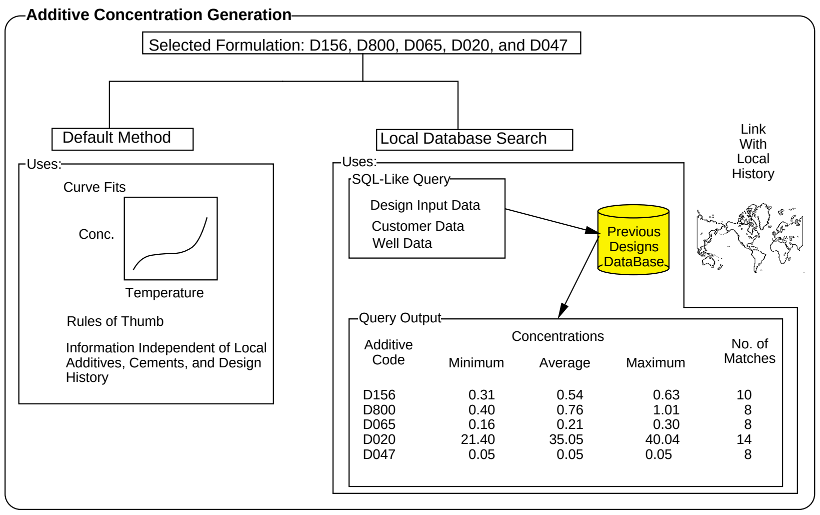
Figure 6. Each Intermediate Solution Node Contains the List of Selected Additives to This Point in the Inference and the Total Confidence Level for the Node. Explanation and warning slots contain a list of codes, the first for the type of solution the node represents and those following for each selected additive. The role of each selected additive is also represented. Node names are indicative of the type of additive most recently selected, with the node number showing the order in which this node was created.

additives with traditional database search techniques that is making SLURRYMINDER an outstanding technical success for the company.

Application Use And Payoff

SLURRYMINDER was officially released to approximately 155 field locations in 55 countries in October 1991. The system was designed for two distinct types of users: sales engineers and laboratory technicians. Sales engineers interact with clients to prepare proposals and bids for services and are usually initially interested only in approximate slurry formulations; fine tuning of the bid occurs after the laboratory technicians test and tune the slurry to obtain the exact quantities of the chemical additives needed to perform the cementing service. Laboratory technicians, however, are interested in obtaining an accurate formulation for two reasons: They want to (1) lower operating costs by reducing the number of expensive testing iterations they must perform and (2) increase their throughput. Both types of user are able to accomplish their goals using SLURRYMINDER.

Although SLURRYMINDER has only been released to our field opera-


Additive Concentration Generation
Selected Formulation: D156, D800, D065, D020, and D047
Default Method
Local Database Search
Uses:
Curve Fits
Conc.
Temperature
Rules of Thumb
Information Independent of Local Additives, Cements, and Design History
Uses:
SQL-Like Query
Design Input Data
Customer Data
Well Data
Previous Designs DataBase
Link With Local History
Query Output

Additive Code	Concentrations Minimum	Average	Maximum	No. of Matches
D156	0.31	0.54	0.63	10
D800	0.40	0.76	1.01	8
D065	0.16	0.21	0.30	8
D020	21.40	35.05	40.04	14
D047	0.05	0.05	0.05	8

Figure 7. Additive Concentrations Can Be Estimated from Two Sources: The Local Database of Previous Designs or a Default Method.
Using the local database provides a link with local design practices and implicitly accounts for local additive and cement variations. Using the default method can be likened to reviewing a general cementing manual for recommendations, where one can find curve fits of data, rules of thumb, and concentration information that is independent of local design history and practice.

tions for a few months, it is already influencing the company's approach to its cementing business. Our European region (which provided a SLURRYMINDER beta testing location) has begun standardizing on SLURRYMINDER to assure that all laboratory personnel use this tool to perform their daily slurry designs. A specially prepared database containing over 300 tests from all parts of the region was prepared and distributed to all European locations for use with SLURRYMINDER. This region is currently delivering cement slurry designs to their customers that incorporate a uniform design philosophy and a common historical support base throughout the entire region, which includes the North Sea and the former Soviet Bloc countries. Northern Africa locations are also reporting good acceptance and design success using SLURRYMINDER.

The company maintains two training centers, one in Kellyville, Oklahoma, and the other in Nottinghamshire, England. These centers are responsible for training all new field employees and updating the skill level of the current field engineers. SLURRYMINDER training has become part of the standard training program in both centers. After several years of this one-time training, a significant portion of our design engineers and technicians will have been trained to use SLURRYMINDER, resulting in its becoming an integral part of the corporate technical structure. Thus, repetitive training costs are reduced, and geographically remote locations gain a technical marketing edge.

Expected Benefits

On a global basis, we perform approximately 36,000 well-cementation jobs annually and design approximately twice this many. For each service recommendation made by a sales engineer, we estimate that personnel savings of 1/2 to 1-1/2 person-hours will be achieved, as follows: Sales engineers typically provide design data to laboratory personnel and request that the laboratory prepare an estimated slurry formulation, without laboratory testing for bidding purposes. This process usually requires approximately 1/2 to 1-1/2 hours for each recommendation from the sales engineer and the laboratory technician. With SLURRYMINDER, the sales engineer can obtain a good initial slurry formulation in less than three minutes without requiring interaction with laboratory personnel. With a conservative basis of 12,000 formulations designed annually using SLURRYMINDER, we have an annual savings of 6,000 to 18,000 person-hours, or 3 to 9 person-years. Intangible benefits, such as better slurry designs using updated technology, which cannot be measured precisely, must also be considered.

Laboratory technicians typically perform an average of four test runs for each slurry pumped. One of our cementing specialists estimated that by using SLURRYMINDER, this number will be reduced by at least one-fourth. Normally, each laboratory test requires approximately four to six hours of expensive machine time to run. If SLURRYMINDER is used to design 12,000 jobs annually, we save worldwide approximately 60,000 machine-hours. Internal accounting audits indicate that replacement parts for the machinery required to run these tests averages about US$10 for each test run; hence, savings resulting from lower machine maintenance costs are estimated at US$120,000 annually.

Application Development and Deployment

In its current form, SLURRYMINDER is the result of a three-year effort in software and knowledge engineering. Generating the knowledge bases represents only approximately 25 to 30 percent of the total effort required to develop the entire application. Currently, we have invested eight person-years in the project.

Development Process

The SLURRYMINDER development process began in February 1989. Two engineers were given the responsibility for performing a feasibility analysis to determine whether a software tool could be developed to help Dowell Schlumberger rationalize and standardize its slurry design efforts on a global basis (figure 8). Earlier efforts at using statistical methods to analyze design data to distinguish definite trends in conjunction with raw material characterization studies had failed to provide the tools and the consistency the company was seeking. Hence, we began looking at expert system techniques to determine if they would be useful in solving our particular problem. Funding for the first year of the SLURRYMINDER project came from a special fund designed to support projects with high risk but high potential return.

As part of the five-month feasibility analysis, approximately 40 individuals in all parts of the company and at all levels were interviewed through a specially prepared questionnaire. These people represented company management, slurry design specialists, laboratory technicians, sales engineers, and some new employees with little slurry design experience. Using the results of the questionnaire and some additional information obtained through follow-up interviews, we analyzed our domain, following suggestions made by Bobrow, Mittal, and Stefik (1986) and using a methodology proposed by Slagle and Wick (1988). Our analysis concluded that the slurry design problem could be solved using expert system technology, and we recommended that we should proceed with prototype development.

The development team was given six months in which to generate a working prototype that would completely design one type of slurry system. User specifications and functional specifications for the prototype were developed in July 1989. We also obtained a portable personal computer on which to run our selected development shell for use while we worked with specialists at locations external to the engineering center. Within our company, cementing specialists are in high demand, and we felt that it would be more economical and convenient if we went to them rather than require them to come to the engineering center.

In December 1989, the prototype was demonstrated to company management and was accepted. Approval for the development version of SLURRYMINDER was obtained in January 1990 and was funded from the general engineering fund like all other engineering projects.

User specifications for the development version, prepared by corporate marketing and engineering personnel, were received in their final form in April 1990, and the functional specifications were ready by May 1990. Much of the knowledge base design was completed during the

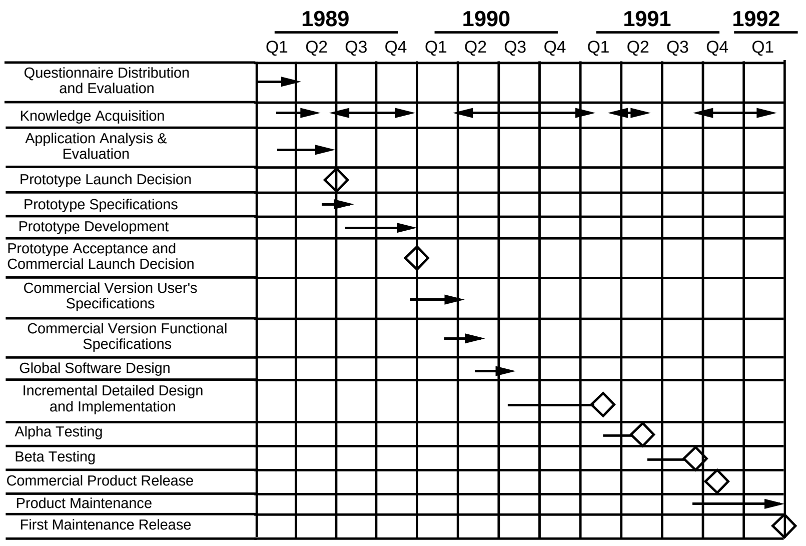


Figure 8. S*LURRY*M*INDER Project Development Timeline.*
Diamonds represent major milestones achieved in the project development process.

prototyping phase; however, we were required to redesign the human interface and incorporate the additional functions into SLURRYMINDER, as required by the user specifications. A global software design was prepared during June and July 1990; however, the detailed design of the software proceeded incrementally with the implementation.

SLURRYMINDER alpha testing began in March 1991 and continued for three months. Our major goal during the alpha test was to ensure that the knowledge within SLURRYMINDER was correct and that it would produce good slurry formulations. As part of the alpha test, we prepared a *knowledge guide*, which is a graphic representation of the design rules within the system, as shown in figures 9 and 10. This knowledge guide can be thought of as a series of multidimensional flowcharts that can be used to follow program control during the reasoning process. Using the knowledge guide, along with the SLURRYMINDER software, allowed us to obtain 100-percent coverage of the system rules during alpha testing. Incidentally, approximately 90 percent of the SLURRYMINDER rules were modified as a result of this process, some significantly.

Beta testing began in June 1991 at three locations external to the engineering center: Africa Regional Office in Paris, France; European Regional Laboratory in Aberdeen, Scotland; and Gulf Coast Divisional Laboratory in New Orleans, Louisiana. The main goals of this testing were to ensure that SLURRYMINDER was robust in a field environment and

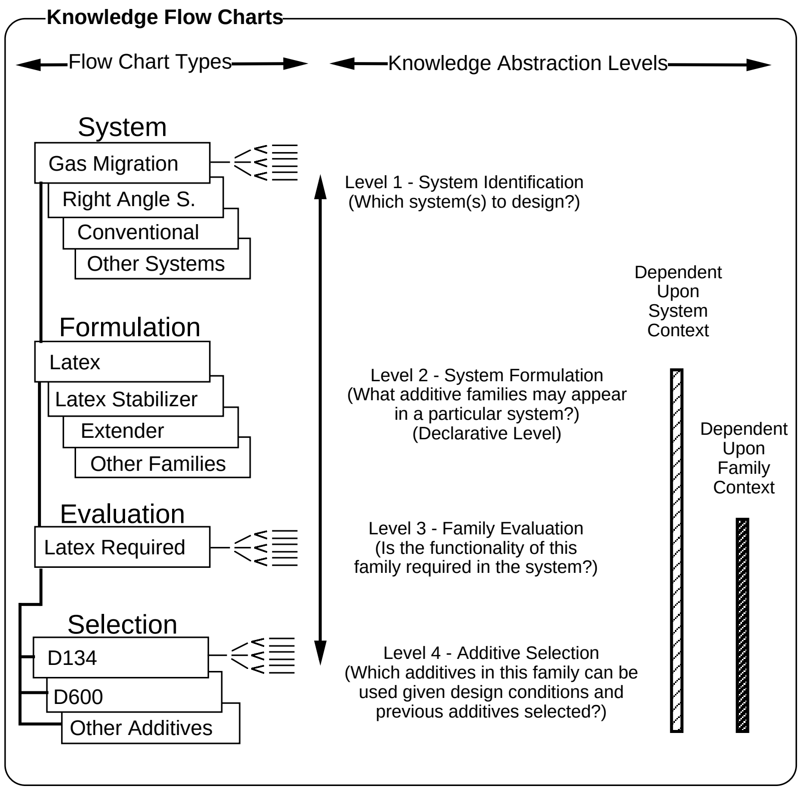

Knowledge Flow Charts
Flow Chart Types
Knowledge Abstraction Levels
System
Gas Migration
Right Angle S.
Conventional
Other Systems
Formulation
Latex
Latex Stabilizer
Extender
Other Families
Evaluation
Latex Required
Selection
D134
D600
Other Additives
Level 1 - System Identification
(Which system(s) to design?)
Level 2 - System Formulation
(What additive families may appear in a particular system?)
(Declarative Level)
Level 3 - Family Evaluation
(Is the functionality of this family required in the system?)
Level 4 - Additive Selection
(Which additives in this family can be used given design conditions and previous additives selected?)
Dependent Upon System Context
Dependent Upon Family Context

Figure 9. The Structure of SLURRYMINDER's Multidimensional Flowcharts. Prepared for experts and novices to follow SLURRYMINDER's reasoning path, each box represents an individual flowchart illustrating the conditions required to prove a particular hypothesis. The formulation level is declarative knowledge and has no explicit validation rules. The structure of these knowledge charts was designed to reproduce the knowledge-abstraction levels used in designing the knowledge representation scheme. Chart content for levels 2 to 4 is dependent on what slurry type was selected at level 1 and how the additive family is used within the selected slurry type.

that the human interface was acceptable to real users. Excellent feedback was obtained during this testing period that resulted in some significant modifications to the human interface and the mechanism for generating additive concentrations when searching the local database. SLURRYMINDER beta testing was completed and accepted in August 1991, with management approving SLURRYMINDER for worldwide deployment.

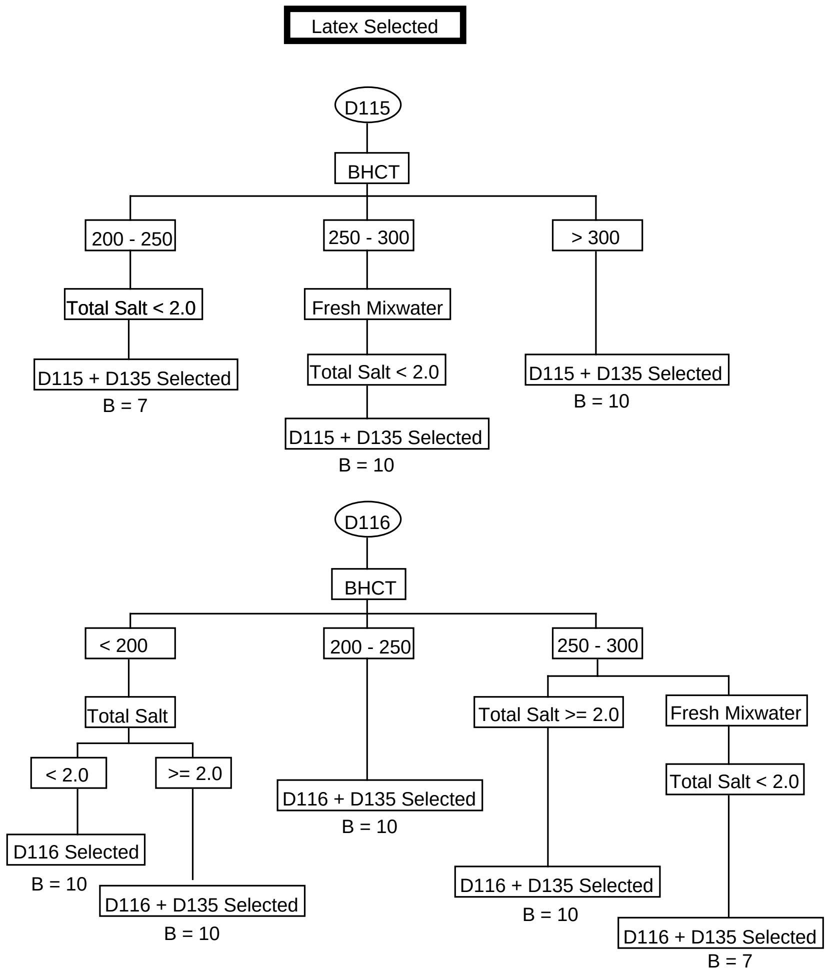


Figure 10. A Typical Knowledge Chart within SLURRYMINDER at Abstraction Level 4, Showing Detailed Conditions for Individual Additive Selection.

Deployment

In September 1991, the SLURRYMINDER executable image and accompanying data files were submitted to our corporate baseline management team in Tulsa, Oklahoma. This team is responsible for preparing a new corporate computer baseline every six months and distributing the software therein to all field locations worldwide. Because this mechanism was already in place for previously existing software, deployment to our field locations was straightforward and consisted of copying the

SLURRYMINDER files into the software baseline, testing the executable file to ensure that it performed properly in the baseline environment, and distributing a tape.

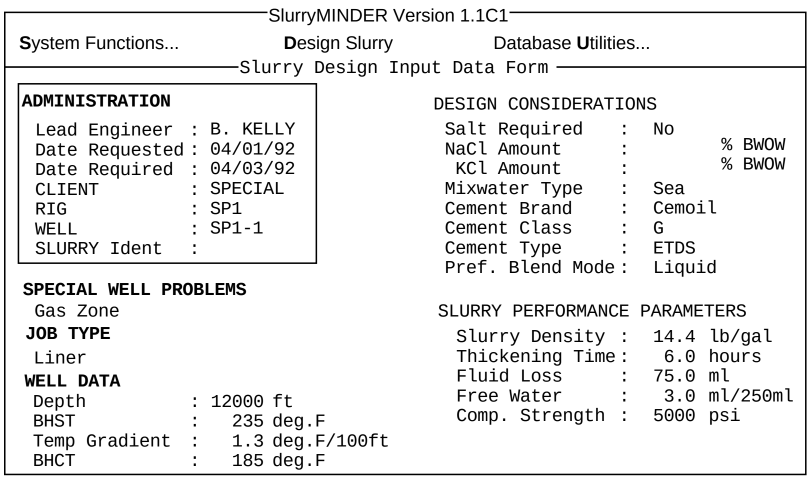
The user's manual for SLURRYMINDER was prepared using LATEX, and the corresponding printable file is contained in the documentation section of the software baseline. At any time, users in any location can obtain the latest documentation for any product in the software baseline by printing the file on their local printer.

By design, SLURRYMINDER is reasonably easy to use even for novice computer users. However, four train-the-trainer training sessions have been held; in Tulsa, Oklahoma; Aberdeen, Scotland; Nottinghamshire, England; and St. Etienne, France. Attendees at these training sessions included regional baseline managers, management personnel, trainers from our two training centers, sales engineers, and laboratory personnel. Because of the geographically diverse nature of our company, each of the individuals attending these training sessions was given the responsibility for training users in their respective regions or locations.

Product Maintenance

Within our company, software product maintenance is performed by members of the software-sustaining section at the engineering center where the product was developed. SLURRYMINDER is now being maintained by a software engineer who was not a member of the original development team. As part of the development effort and good software engineering practice, the original development team created two documents to aid in the maintenance of the SLURRYMINDER software: the SLURRYMINDER Knowledge Base Maintenance Guide and the SLURRYMINDER Software Maintenance Guide. These complementary documents respectively describe the internal logic and software structures of the knowledge bases and the rest of the software in sufficient detail to facilitate maintenance. Coupled with the Knowledge Guide (knowledge flowcharts) and the product specification documents, maintainers have significant maintenance resources available to them.

Cement slurry design is an evolving domain; we have already begun modifying the knowledge bases by adding recently developed cementing additive products and a new slurry formulation technology to SLURRYMINDER. It appears that the architecture of the knowledge bases is well suited to domain evolution because these modifications were made by an individual who was not on the original development team. New releases of SLURRYMINDER will occur every six months as the company's software baseline is updated and redistributed to all field locations.



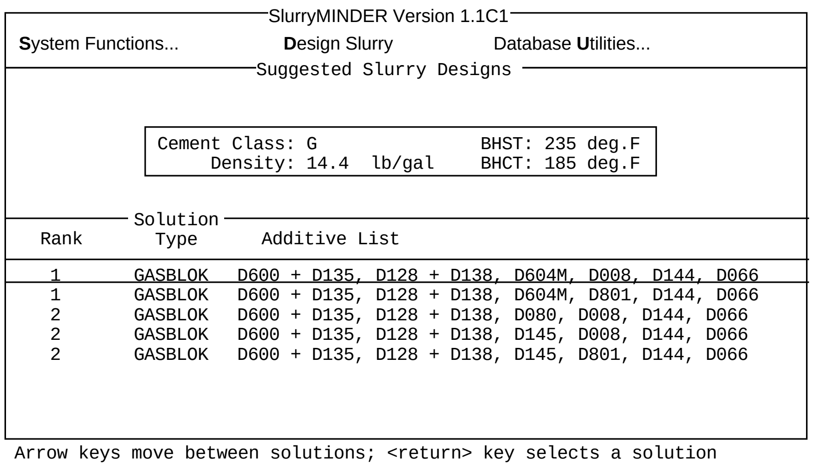
```
Slurry Design Input Data Form

ADMINISTRATION                              DESIGN CONSIDERATIONS
 Lead Engineer  : B. KELLY                   Salt Required    :  No
 Date Requested : 04/01/92                   NaCl Amount      :        % BWOW
 Date Required  : 04/03/92                    KCl Amount      :        % BWOW
 CLIENT         : SPECIAL                    Mixwater Type    :  Sea
 RIG            : SP1                        Cement Brand     :  Cemoil
 WELL           : SP1-1                      Cement Class     :  G
 SLURRY Ident   :                            Cement Type      :  ETDS
                                             Pref. Blend Mode :  Liquid
SPECIAL WELL PROBLEMS
 Gas Zone                                   SLURRY PERFORMANCE PARAMETERS
JOB TYPE                                      Slurry Density :  14.4 lb/gal
 Liner                                        Thickening Time:   6.0 hours
WELL DATA                                     Fluid Loss     :  75.0 ml
 Depth          : 12000 ft                    Free Water     :   3.0 ml/250ml
 BHST           :   235 deg.F                 Comp. Strength :  5000 psi
 Temp Gradient  :   1.3 deg.F/100ft
 BHCT           :   185 deg.F
```

Figure 11. SLURRYMINDER's Primary Data Input Form.
Input data consist of well data such as depth, temperatures, and special downhole conditions; target slurry performance parameters such as density and thickening time; properties of the cement; and administration data such as well logistical information.

An Example Using SLURRYMINDER

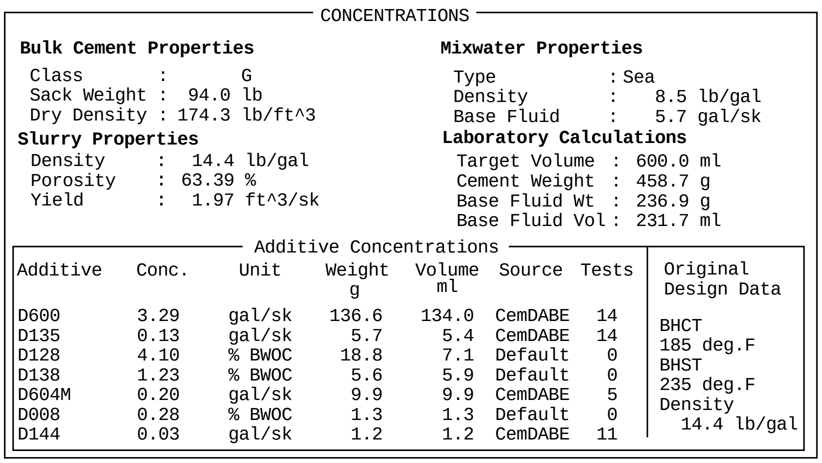
A typical use of the SLURRYMINDER system is illustrated by the following problem: A client is drilling an offshore oil well at 12,000 feet below the surface. The temperature in the underground formation is approximately 235 ˚F, and the slurry density must be around 14.4 pounds to a gallon to maintain the hydrostatic head and keep formation fluids from entering the well bore. Because the rig is offshore, seawater will be mixed with the cement powder to create the cement slurry, and liquid chemical additives are preferred over solids because of limited bulk handling and storage facilities on the rig. The cement slurry must not set up for at least six hours to allow proper placement around the cement casing. This particular cement system must also isolate an adjoining formation with significant potential for gas to enter the cement matrix and corrupt the cement, allowing gas to migrate up the casing column and create a potentially dangerous situation. In a well this deep, class G well cement will be used; the particular brand available on location is easy to disperse in the presence of the salt in the seawater. The slurry should have low free water and fluid loss and should achieve a good compressive strength in 24 hours.

Design data of this nature are entered into the SLURRYMINDER input



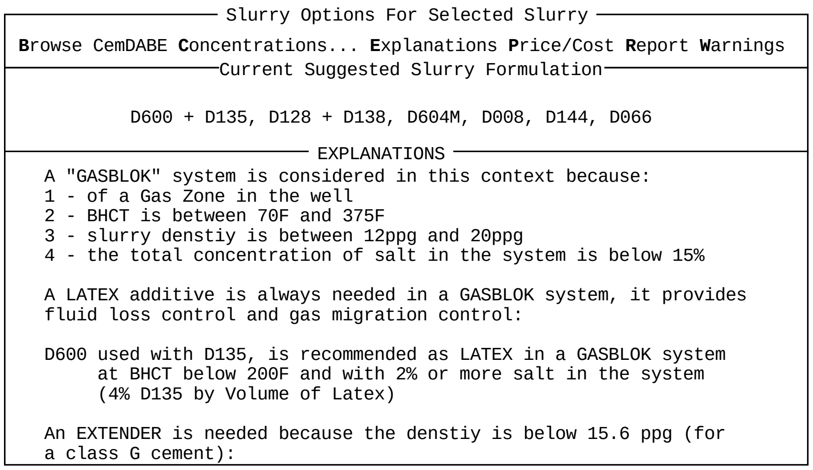
```
                         SlurryMINDER Version 1.1C1
 System Functions...        Design Slurry        Database Utilities...
                          Suggested Slurry Designs

            Cement Class: G               BHST: 235 deg.F
                 Density: 14.4  lb/gal    BHCT: 185 deg.F

                 Solution
   Rank            Type      Additive List

    1       GASBLOK   D600 + D135, D128 + D138, D604M, D008, D144, D066
    1       GASBLOK   D600 + D135, D128 + D138, D604M, D801, D144, D066
    2       GASBLOK   D600 + D135, D128 + D138, D080, D008, D144, D066
    2       GASBLOK   D600 + D135, D128 + D138, D145, D008, D144, D066
    2       GASBLOK   D600 + D135, D128 + D138, D145, D801, D144, D066

 Arrow keys move between solutions; <return> key selects a solution
```

Figure 12. An Example of Slurry Additive Formulations Recommended by SLURRYMINDER.
Formulations include the list of chemical additives and the rank of each formulation. By default, the highest-ranking slurry is highlighted for further processing to obtain additive concentrations, explanations, warnings, pricing information, or a design report.



```
                              CONCENTRATIONS
 Bulk Cement Properties                      Mixwater Properties
  Class       :       G                       Type          : Sea
  Sack Weight :  94.0 lb                      Density       :   8.5 lb/gal
  Dry Density : 174.3 lb/ft^3                 Base Fluid    :   5.7 gal/sk
 Slurry Properties                           Laboratory Calculations
  Density     :  14.4 lb/gal                  Target Volume  : 600.0 ml
  Porosity    : 63.39 %                       Cement Weight  : 458.7 g
  Yield       :  1.97 ft^3/sk                 Base Fluid Wt  : 236.9 g
                                              Base Fluid Vol : 231.7 ml
                        Additive Concentrations
 Additive   Conc.    Unit    Weight  Volume  Source  Tests   Original
                               g       ml                    Design Data
 D600       3.29    gal/sk    136.6   134.0  CemDABE   14    BHCT
 D135       0.13    gal/sk      5.7     5.4  CemDABE   14    185 deg.F
 D128       4.10    % BWOC     18.8     7.1  Default    0    BHST
 D138       1.23    % BWOC      5.6     5.9  Default    0    235 deg.F
 D604M      0.20    gal/sk      9.9     9.9  CemDABE    5    Density
 D008       0.28    % BWOC      1.3     1.3  Default    0      14.4 lb/gal
 D144       0.03    gal/sk      1.2     1.2  CemDABE   11
```

Figure 13. Additive Concentrations Generated by SLURRYMINDER Are Used to Compute Quantities Required to Prepare the Slurry Formulation in the Laboratory.
Additional calculations are performed to obtain the quantity of water to use and the volume of slurry obtained from one sack of cement.



```
                    Slurry Options For Selected Slurry
Browse CemDABE Concentrations... Explanations Price/Cost Report Warnings
                    Current Suggested Slurry Formulation

          D600 + D135, D128 + D138, D604M, D008, D144, D066
                              EXPLANATIONS
   A "GASBLOK" system is considered in this context because:
   1 - of a Gas Zone in the well
   2 - BHCT is between 70F and 375F
   3 - slurry denstiy is between 12ppg and 20ppg
   4 - the total concentration of salt in the system is below 15%

   A LATEX additive is always needed in a GASBLOK system, it provides
   fluid loss control and gas migration control:

   D600 used with D135, is recommended as LATEX in a GASBLOK system
        at BHCT below 200F and with 2% or more salt in the system
        (4% D135 by Volume of Latex)

   An EXTENDER is needed because the denstiy is below 15.6 ppg (for
   a class G cement):
```

Figure 14. An Example of the Type of Explanations Generated during a Session with SLURRYMINDER.
Each explanation includes information on the criteria used to select what type of slurry to design, why a particular additive family is necessary in the formulation, and what the selection conditions are for each individual additive.

data form, as illustrated in figure 11. A series of data checks designed into the intelligent interface ensure that data entered in the fields are internally consistent. Prior to invoking the inference engine, users can specify the maximum number of solutions they would like to obtain; the default is five. This number is used internally during inferencing to prune branches from the search space; only the top candidate solutions are kept for subsequent inferencing.

We intentionally designed the inferencing mechanism to be a black box as far as users are concerned. Input data are volunteered to the inference engine, and little interaction with the user is required until the inference is complete, and the solutions have been generated.

Figure 12 illustrates the output obtained by inferencing using the data from figure 11. Five solutions were requested, with two of the solutions ranked equally as the best. Careful examination reveals that only subtle differences exist between the different solutions in this example; however, the different internal chemical additive codes have great significance to the field personnel in terms of how well these additives perform in different circumstances. Additives joined by a plus sign indicate that the first additive requires the aid of the second to provide the required functions in the slurry formulation.

After selecting one of the formulations for further processing, users have the option of generating additive concentrations, viewing explanations or warnings, generating a design report, or obtaining pricing information for the particular formulation. Figure 13 illustrates the results obtained from invoking the concentration generation mechanism, and figure 14 shows the initial explanation output screen obtained from the explanation and warning subsystem.

Acknowledgments

We would like to thank the many individuals within the Dowell Schlumberger organization for their time and efforts, so freely given to make this project a success. We also want to express our appreciation to Reid Smith of the Schlumberger Laboratory for Computer Science for reviewing drafts of this manuscript and providing insightful suggestions.

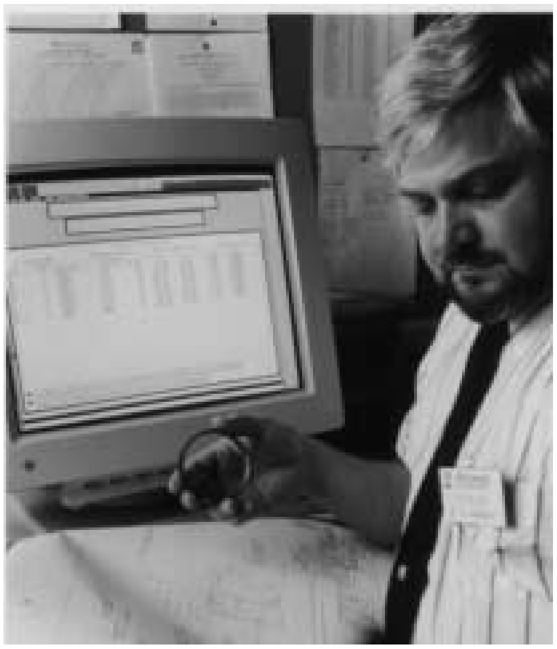
References

Bobrow, D. G.; Mittal, S.; and Stefik, M. J. 1986. Expert Systems: Perils and Promise. *Communications of the ACM* 29(9): 880–894.

Nelson, E. 1990. Well Cementing. Houston, Texas: Schlumberger Educational Services.

Slagle, J. R., and Wick, M. R. 1988. A Method for Evaluating Candidate Expert System Applications. *AI Magazine* 9(4): 44–53.

Wolsfelt, G. C.; Roger, C.; and Fenoul, R. 1989. A Cementing Job Preparation Advisor System. Presented at the Sixty-Fourth Annual Technical Conference and Exhibition of the Society of Petroleum Engineers, San Antonio, Tex., 8–11 October.



SLURRYMINDER