

Automating Forms Publishing with the Intelligent Filing Manager^[1]

K. C. Mukherjee

CCH-LIS, Wolters-Kluwer US

1633 Broadway, 30th floor

New York, NY 10019, U.S.A.

mukherjeek@cch-lis.com

ABSTRACT

This paper describes how logic programming provides flexibility in complex forms filings not previously available with hard-coded solutions. The Intelligent Filing Manager (INTELLIFM) software, a logic programming application, has been developed for high-volume forms publishing applications where large sets of varied forms need to be filed simultaneously. In the United States, for large insurers who wish to do business in every state, there are approximately 700 different forms. Logic programming, combined with shrink-wrapped applications such as Microsoft Excel, is used to automate such large-scale forms filings that are subject to changes in government regulations. The INTELLIFM system combines Artificial Intelligence (AI), Graphical User Interface (GUI) and Databases with Portable Document Format (PDF) technology. The filing manager software has a built-in, intelligent question-answering (Q&A) module. INTELLIFM represents a system in which AI and dynamic GUI generation play powerful roles, and automate forms publishing in a novel way.

1. INTRODUCTION

Today, logic programming successfully automates legal filings in the insurance industry. Regulatory requirements, annual reporting, rates, policies, approval for underwriting and the oversight of sales personnel make the insurance industry the third most heavily regulated in the United States. As mentioned earlier, for large insurers doing business in every state, there are approximately 700 different types of forms. This enormous multi-jurisdictional forms filing project is an example of a market previously ignored because of the variations in forms.

The demand for automated forms publishing exists in other industries as well. In the United States, FDA regulatory requirements for food and drug manufacturers require an enormous amount of forms processing as does the Bureau of Alcohol, Tobacco and Firearms. Doctors must process large numbers of insurance forms for their patients. In the public sector, State and Federal departments of social services are also burdened with forms. At social service offices, employees who process applications for mothers applying for aid to families with dependent children must learn to fill out 79 different forms.

Although the number of forms varies from client to client, there may be as many as 36 forms per applicant. Each of the forms may contain redundant fields such for name, address, social security and other common data.

Compliance requirements change frequently, in forms publishing, making it expensive and time consuming to re-write, test and debug code for the lifetime of a Filing Manager. The INTELLIFM system uses a simple and well-designed document model that provides the necessary flexibility. The Q&A software is generic, easily maintainable and extensible. The Q&A knowledge base contains rules that will be executed, leaving the filing manager software uncluttered with complex, procedural details.

The INTELLIFM GUI is generated dynamically from database specifications, and is controlled through high-level rules. Rules enable and disable screen controls, remove redundancy in question requests, set control values and replicate answers on form fields. Domain experts, who may be non-technical, write these rules.

For multi-jurisdictional filings or filings in many states, INTELLIFM always computes the minimal sets of questions. The expert system's strength is derived from the simplistic way the knowledge base is created with off-the-shelf software, and the flexibility in handling changes in forms requirements.

2. ANALYSIS OF DATA AND RULES

The task of automating insurance form filings is reverse engineered from the system's final output - the PDF file. A PDF file preserves the dimensions and layout of a form for viewing and printing.

To start, the scanning of a paper form generates a PDF image. The PDF fields are assigned unique names using Adobe's Acrobat Exchange software. Because the INTELLIFM software supports multi-jurisdictional filings, the expression "form-set" is used to describe a collection of fielded PDF forms. The process of generating the prompting questions used to add data to form fields is accomplished through the GUI. Prompts are associated with fields, and fields are specified with controls that need to be displayed in a Q&A session. Controls are of different types - such as edit-box, list-box, check-box, radio-button, spin control

[1] An earlier version of this paper appeared in the ICAIL-99 conference proceedings

or grid. Often, related controls are grouped together.

Grouping And Ordering Prompts

Most work in forms publishing automation is accomplished by domain experts. Their understanding of compliance requirements help to organize or cluster questions that belong together on the GUI. For a form based Q&A, prompts are classified in two categories – form independent and form dependent. Form independent prompts define a transaction - their answers determine the purpose of a filing. The answers to form dependent prompts populate the fields appearing in printed copies of the forms. For notational convenience, all form independent prompts start with the letter T (signifying transaction definition) while all form dependent prompts start with the letter Q (signifying form level question).

Prompts need to be ordered. A person's name has to be asked before the date of birth, or the filing type has to be asked before the jurisdiction. So, priorities are assigned to prompts. T prompts, with priority code 1, are displayed at the very beginning of a Q&A session. Q prompts, with priority codes of 2 or greater, are fetched after processing answers to the T prompts. Within a group, member prompts are alphabetically ordered. For example, prompts Q1a through Q1d have equal numeric priority of 2, and are displayed in the alphabetic sequence – Q1a, Q1b, Q1c and Q1d (see Table 1).

Table 1: PROMPT-FIELD spreadsheet

PROMPT	PRIORITY	PDF FIELD
Q1	2	GroupAgentName
Q1a	2	AgentFirstName
Q1b	2	AgentMiddleName
Q1c	2	AgentLastName
Q1d	2	AgentSuffix
Q2	4	GroupSpouseName
Q2a	4	SpouseFirstName
Q2b	4	SpouseMiddleName
Q2c	4	SpouseLastName
Q3	3	GroupMaritalStatus
Q3a	3	StatusSingle
Q3b	3	StatusMarried
Q4	6	GroupAppDetails
Q4a	6	LetterClearance
Q4b	6	LetterCertification
Q4c	6	ProofNasd
Q4d	6	ProofProf
Q4e	6	Other
Q4f	6	Specify
Q5	5	DBAName

With this prioritization scheme, the prompts are sorted in ascending order using a compound key (Priority-Code and Prompt-Id) so that that they are always displayed in a logical sequence.

Specifying Data

Let's consider the example of obtaining an "Initial License" (Filing Type code is LIC) for a producer allowed to do business as a "Non Resident" (Residency Status code is NON) in the states of Pennsylvania (PA) and Rhode Island (RI). The producer is an "Individual" (Producer Type code is IND), seeking an "Agent and All Other" license (License Type code is AGT), and planning to sell insurance for "All Lines" (Categories of Lines of Authority code is ALL).

For a typical insurance filing, examples of transaction definition questions are:

T1 ("Select a Filing Type") with options T1a ("Initial License"), T1b ("License Amendment"), T1c ("License Reinstatement"), T1d ("License Termination"), T1e ("Appointment") and T1f ("Appointment Termination").

T2 ("What is the applicant's residency status?") with options T2a ("Resident") and T2b ("Non-Resident").

T3 ("Select a jurisdiction") with the option to select any one of the 51 states (including Washington DC).

T4 ("Select one or more jurisdictions") with the option to select one or more of the 51 states.

T5 ("How many producers are being appointed or terminated?") accepting numeric values.

T6 ("Select the producer type") with options T6a ("Individual"), T6b ("Individual in a firm"), T6c ("Firm") and T6d ("Bank").

T7 ("License Type") with options T7a ("Trainee"), T7b ("Solicitor"), T7c ("California Limited Liability Company") etc.

T8 ("Categories of Lines Of Authority") with options T8a ("Property, Casualty, Surety or Marine"), T8b ("Fire & Casualty"), T8c ("Life & Health") etc.

Answers to prompts T1 through T8 are used to compute the set of Q prompts that get displayed in a Q&A session.

The transaction codes are specified in Excel. In our example, the choices generate two form-sets – PA1 and RI1 (see Table 2). Let's assume that PA1 and RI1 consist of forms - PA001.pdf and RI001.pdf respectively.

Table 2. TRANSACTION-FORMSET spreadsheet

FILING TYPE	RES TYPE	PROD TYPE	LIC TYPE	LOA CAT	JURIS	FORM SET
LIC	NON	IND	AGT	ALL	PA	PA1
LIC	NON	IND	AGT	ALL	RI	RI1

Prompts are defined for fields in PA001.pdf and RI001.pdf forms (see Table 3).

Going back to Table 1, we can see how the prompt priorities are defined. Further, the FIELD column in that table specifies the form fields (as defined in the PDF images) corresponding to the prompts.

Table 3. FORM-PROMPT spreadsheet

FORM	PROMPT	RULE_PRECONDITION
PA001	Q1	
PA001	Q1a	
PA001	Q1b	
PA001	Q1c	
PA001	Q1d	
PA001	Q2	do_ask_question(Q2) :- is_answer(Q3b, yes).
PA001	Q2a	
PA001	Q2b	
PA001	Q2c	
PA001	Q3	
PA001	Q3a	
PA001	Q3b	
PA001	Q4	
PA001	Q4a	
PA001	Q4b	
PA001	Q4c	
PA001	Q4d	
PA001	Q4e	
PA001	Q4f	do_ask_question(Q4f) :- is_answer(Q4e, yes).
RI001	Q1	
RI001	Q1a	
RI001	Q1b	
RI001	Q1c	
RI001	Q1d	
RI001	Q2	do_ask_question(Q2) :- is_answer(Q3b, yes).
RI001	Q2a	
RI001	Q2b	
RI001	Q2c	
RI001	Q3	
RI001	Q3a	
RI001	Q3b	
RI001	Q5	

Like an HTML browser, INTELLIFM parses tags and generates controls dynamically. Special tags are defined to render a rich array of controls. A TextGroup specifies a collection of edit boxes. Similarly, RadioGroup and CheckGroup specify groups

of radio buttons and check boxes respectively. In order to display groups of questions within a Q&A session, combinations of controls may be specified.

For example, GroupAgentName is a TextGroup with AgentFirstName, AgentMiddleName, AgentLastName and AgentSuffix as member controls (see Table 4). Similarly, a CheckTextGroup has check and edit boxes grouped together into a composite question. When rendered on screen, the member controls of a group are enclosed in a rectangular frame with the common group level caption (eg: "Please check all items that are being submitted with the Pennsylvania application:").

Masks or formatting codes (eg: UpperFirst for initial capitals), maximum number of characters that can fit into a field (eg: 30 for first name, 5 for name suffix) and other properties may be specified for individual controls.

Specifying Rules

In the INTELLIFM system, the GUI is programmed to respond to events using high-level rules. The predicates are classified into two categories - *action* and *state observation*.

The action predicates specify what needs to be done. Typically, they appear as the heads of rules. An example is the *do_ask_question* predicate. It specifies whether a question needs to be asked during a Q&A session. If true, the question's control gets enabled.

On the other hand, the state observation predicates specify preconditions. They appear as rule bodies, and are used for examining facts at a particular instance of time. An example is the *is_answer* predicate that checks for prompt-answer combinations.

The Q&A rules are entered in the RULE_PRECONDITION column of the FORM-PROMPT spreadsheet (see Table 3). Domain experts enter the rules using a well-defined set of predicates.

An offline Database and Rule-base Generator (DRG) utility (see Figure 1) processes the Excel workbook.

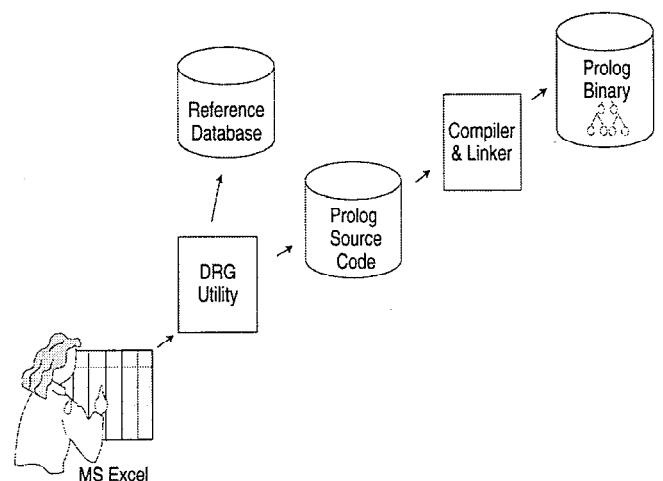


Figure 1. Database and Rule-base Generator (DRG) utility

Table 4: FIELD spreadsheet

FIELD	GROUP	CTRL TYPE	CTRL CAPTION	CTRL FORMAT	CTRL SIZE
GroupAgentName		TextGroup	What is the name of the applicant?		
AgentFirstName	GroupAgentName	Text	First Name:	UpperFirst	30
AgentMiddleName	GroupAgentName	Text	Middle Name:	UpperFirst	30
AgentLastName	GroupAgentName	Text	Last Name:	UpperFirst	30
AgentSuffix	GroupAgentName	Text	Suffix:	UpperFirst	5
GroupSpouseName		TextGroup	What is the name of the applicant's spouse?		
SpouseFirstName	GroupSpouseName	Text	First Name:	UpperFirst	30
SpouseMiddleName	GroupSpouseName	Text	Middle Name:	UpperFirst	30
SpouseLastName	GroupSpouseName	Text	Last Name:	UpperFirst	30
GroupMaritalStatus		RadioGroup	What is the applicant's marital status?		
StatusSingle	GroupMaritalStatus	Radio	Single:		
StatusMarried	GroupMaritalStatus	Radio	Married:		
GroupAppDetails		CheckTextGroup	Please check all items that are being submitted with the Pennsylvania application:		
LetterClearance	GroupAppDetails	Check	Letter of Clearance		
LetterCertification	GroupAppDetails	Check	Letter of Certification		
ProofNasd	GroupAppDetails	Check	Proof of NASD		
ProofProf	GroupAppDetails	Check	Proof of Professional Designation		
Other	GroupAppDetails	Check	Other		
Specify	GroupAppDetails	Text	Please Specify:		
DBAName		Text	Enter the applicant's DBA name (doing business as) used in Rhode Island:	UpperFirst	100

Workbook processing is done in phases. In the first phase, the DRG utility generates the reference database. In doing so, the utility checks for referential integrity and creates relationships among the different entities – TRANSACTION, FORMSET, FORM, PROMPT, FIELD and CONTROL.

In its second phase, DRG extracts Q&A rules, parses them, checks for syntax errors and missing identifiers. Prompts and fields appearing as predicate arguments are validated against the reference database.

Additionally, the DRG utility intelligently edits business rules entered in Excel, and makes them conform to the ISO Prolog syntax - all prompts and fields appearing as arguments of rules are converted to lower-case by the utility.

For example, as per insurance regulations - a producer can hold a resident license of only one state, but non-resident licenses for many states. In Excel, the rules are entered as:

```
do_ask_question(T3) :- is_answer(T2a, yes).
do_ask_question(T4) :- is_answer(T2b, yes).
```

These rules are translated by DRG to become:

```
do_ask_question(t3) :- is_answer(t2a, yes).
do_ask_question(t4) :- is_answer(t2b, yes).
```

After the Prolog source files are generated by the DRG, they are then compiled and linked (see Figure 1).

Now, let's look at an example of a form level rule for insurance. After being processed, the rule appears in the Prolog source as:

```
%-----[Statement from Excel Row 8757.01]-----%
do_ask_question(q1012) :-
    is_selected_form('LAAPPBCZVA'),
    (is_answer(t5, X), X = 1).
```

The rule specifies: ask Q1012 - "What is the applicant's resident state?" - only when doing an "Appointment" (APP) filing for a single producer in the state of Louisiana (LA) where the producer is authorized to sell "Variable" (VA) type of insurance. The form is used for both (B) resident and non-resident filings, and for a common (C) producer type and a

generic (Z) license type. The *is_selected_form* predication ensures that Q1012 gets asked only if LAAPPBCZVA is generated by the filing. Further, from a business perspective, an "Appointment" filing can be done for multiple producers in the state of Louisiana. So, the spin control for prompt T5 gets enabled in the Q&A. However, as is clear from the wording of the question, Q1012 is not applicable for multiple producers and should not be asked if T5 is set to more than one. Hence, T5's answer is checked in the rule body to ensure that the edit box control for Q1012 gets enabled only when the number of producers is set to one.

3. FILING MANAGER COMPONENTS

INTELLIFM has four on-line COM components – UIG (User Interface Generator), SQL (Structured Query Language), EPS Encapsulated Prolog Services) and P&P (Persistence and Printing).

INTELLIFM is an expert system built in Prolog with a C++ application superstructure using Microsoft's Component Object Model (COM). The components interact as shown (see Figure 2). The SQL and EPS components work off the reference database and compiled Prolog rules binary file that are generated from the contents of the Excel workbook.

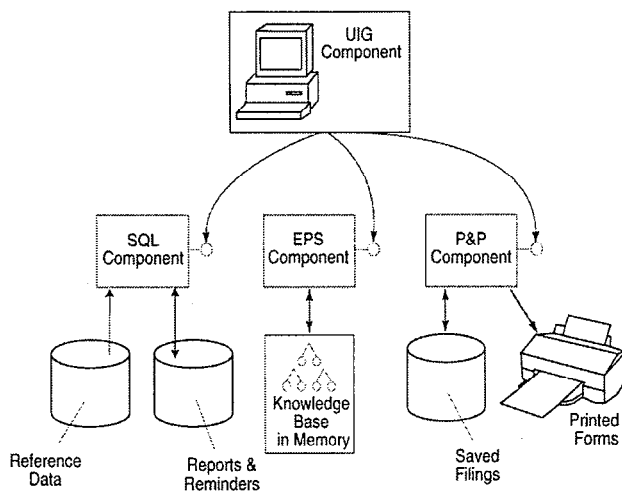


Figure 2: INTELLIFM Components

On invoking INTELLIFM, the UIG component is launched first. UIG creates instances of the other components in memory. It then interacts with the SQL component and displays the form independent (transaction definition or T) prompts. After specifying the transaction choices, when the user begins a filing questionnaire, UIG transmits the user's selections to the SQL component. The SQL component then fetches all form level questions or Q prompts, eliminates duplicates, sorts the prompts and computes the minimal set of questions for the Q&A session. Along with the prompts, the SQL component also fetches the associated PDF field names and the GUI control specifications.

An array of COM structures is marshaled back to the UIG component. UIG parses the structures to dynamically create user interface controls and display the Q&A screens. The front-end

has the all too familiar wizard like appearance. It sports Back-Next buttons and leads the user from page to page with logically arranged questions.

Besides, the SQL component maintains a Reports and Reminders database to track miscellaneous information used for generating reports on previously executed filings.

The EPS component initializes the Prolog inference engine and loads the knowledge base in memory. If we conceptualize the INTELLIFM system as a graphical robot, then the GUI controls are its body, and Prolog rules are its brains. When the user types on the keyboard or clicks with the mouse to answer questions, the UIG component communicates the responses to the EPS component. The EPS component then invokes Prolog functions to assert or retract facts, and evaluate preconditions. It advises the UIG component - how to redraw the user interface.

The same array of COM structures organizes between the UIG and EPS components. Based on information communicated to it, the UIG enables and disables controls on the screen, populates controls with user's answers, displays computed results, and does anything else that the content creators may have programmed with rules.

When the user wants to save and open filings, or print and print-preview PDF forms, the UIG component passes relevant information to the P&P component. To save filings, the P&P component persists user's answers as large binary objects in the customer database.

For printing and previewing of forms, the P&P processes answers to generate intermediate FDF (Format Data Format) files, and then invokes Adobe's Acrobat Reader to generate the final PDF images from the corresponding FDF files.

4. DISTRIBUTED ARCHITECTURE

The on-line components are programmed as dynamic link libraries (DLL's) for Microsoft Windows. They communicate using COM interfaces. The components can either be run as a stand-alone application, or distributed over the Internet.

In the stand-alone version, the four on-line COM components, the inference engine and the compiled Prolog rules are installed on the user's machine. The reference database (Microsoft Access) and PDF forms reside locally.

In the Internet version, the reference database (ORACLE) and the PDF forms are located on a central server. The SQL, EPS and P&P components support dual interfaces and are Automation enabled. The SQL and P&P are registered as Microsoft Transaction Server (MTS) components on the server, and are called from Active Server Pages (ASP). The EPS component, the inference engine and the compiled Prolog rules are downloaded. Browser scripts invoke the EPS component to respond to events and redraw HTML controls that are generated dynamically. The users' answers are transmitted over the Internet. The P&P component saves filings and generates completed PDF forms on the server. The users can download their completed forms.

5. CONCLUSION

Major Insurance companies are using *CT Insurance Advantage*. INTELLIFM is being deployed to complete many insurance filings within a short time.

Like the insurance sector, corporate and UCC sectors have to meet regulatory compliance requirements. Corporations perform filings in order to qualify for doing business in a state, to withdraw their business, and to amend the certificates of incorporation and authority. UCC stands for Uniform Commercial Code. UCC is divided into nine articles. Article 9

deals with Secured Transactions, Sales of Account, Contract Rights and Chattel Paper. Customers search and file public records to find out if any of the collateral that they are acquiring, investing or selling is encumbered by liens.

Besides automating corporate and UCC filings, the generic nature of the INTELLIFM system makes it flexible enough to be used for a number of industries including health insurance, food manufacturing and accounting. This knowledge base driven program, combined with a dynamically generated graphical user interface is a Q&A wizard capable of automating tasks as either a stand-alone or a Web-based application.