

# **Automatic Application Generation: Techniques for Rapid Development of Rich-featured Information Systems**

## **Abstract**

The methodologies of software development are, in general, slow, costly and have risks of failure. Developing software, in general, means writing programs. Architects of software development tools are attempting to make it possible to do more work by giving definitions, leaving little for writing code. We propose “Data Furnishing Techniques and Tools”, an attempt to move some steps forward in this direction. The techniques are based on a novel idea of “generating” business applications automatically, with very little input from the developer. Data Furnishing Techniques exploit the power of metadata and semantics, which are initially acquired by analyzing the schema. Additional information about schema objects is provided by the developer. Data Furnishing Tools benefit from the existence of frequently used patterns that are common in most business applications. The tools reduce the development time and cost drastically, and make it possible to add rich features to an information system. This can be particularly useful in emerging situations. Contact: (Najmul Ikram Qazi) najmalikram@yahoo.com.

## **1. Background**

The methodologies of software development are, in general, slow, costly and have risks of failure. Many software development tools and methodologies require us to freeze our specifications at a very early stage. It is becoming increasingly important to reduce the development time, cost and risk of failure. Developing software, in general, means writing programs. Architects of software development tools are attempting to make it possible to do more with less coding. Is it possible to generate software applications automatically? If yes, to what extent? We attempt to move some steps forward in this direction.

We aim to develop common business applications more rapidly and effectively. Most business applications are based on data, organized in a database. A good application should provide the user with a good control over data. This includes forms for data entry and viewing, powerful means to retrieve data, search facility, detailed and summary reports and similar features. It is assumed that the developer will follow some design guidelines, and that the application is data centric. With these assumptions, we can facilitate the developer to incorporate many pre developed data features in all his objects with very little effort.

## **2. Is it possible to make software without writing programs?**

Developing software, in general, means writing programs. The authors of visual tools attempted to reduce the need to write programs and allowed software developers to design screens visually. Rapid Application Development tools offer significant advantages and contribute greatly to the success of software projects.

The idea of Data Furnishing Tools is an attempt to move some steps forward in this direction. The attempt has been quite successful and encouraging. It aims that the developer should need to write very little code, while the rest is “generated”.

Data Furnishing Architecture can be viewed as an extended form of RAD. It asks the developer to qualify his data objects. Then it makes various decisions automatically on the basis of the given definitions. As a result, the need to write code is significantly reduced, and in some cases, eliminated.

### **3. Data Furnishing Techniques**

Data Furnishing Techniques primarily exploit the power of metadata and semantics. The properties of schema objects are initialized to suitable default values. The developer can then specify the properties of tables, views, fields etc. This way, they define their data.

Data Furnishing Techniques use application patterns that are highly generic, and are applicable to diverse business domains. The Data Furnishing patterns were identified from various business applications. These patterns can be combined to compose major part of information system.

With data semantics in hand, each frequently used application pattern (e.g. form) is implemented as a generic tool. How the patterns are selected, and what properties are included in the system, are key factors in the overall effectiveness of the development environment. If the developer is loaded with too many properties to be specified, the development will slow down. If the pattern does not meet the requirements of application, the developer will have to do customized implementation. Effective use of database views is important for the maximum benefit.

### **4. How do the Techniques and the Tools Work?**

Data Furnishing Tools are specifically useful for applications, which depend heavily on data. Examples are accounting applications, human resource management, school or university management, inventory, sales and marketing applications. While designing the application, the general style suggested by Data Furnishing architecture is followed to make maximum use of the features and facilities offered by the tools. After the requirements analysis, a reasonable database schema is built.

Data Furnishing Tools read the schema and initialize the properties of various objects. The developer (or administrator) creates data groups, user groups and some users. Mapping of data groups and user groups is created, and access rights are specified. These configurations can be done directly in the database or through the user interface.

For each CRUD requirement, a database view or Select statement is made. This data source is then configured in DataFurnish as a data object, and few properties are specified. Each such entry corresponds to a menu item that is available to one or many

user groups. No additional set up is required for menu set up. There is no compilation or code generation. The developer can now run the DataFurnish instance and login as a user (maybe administrator). The developer can modify the configurations and instantly see the effects on the application.

## 5. Automatic UI Views Offered by Data Furnishing Tools

When the developer of the application wants to add some option in the application, they generally create a database view, and configure it in DataFurnish. In this process, they specify a UI view for the data object.

Depending upon the UI view specified, DataFurnish automatically provide the required screen. Some of the UI views are discussed below:

### Form View

This UI view is an automatically shaped form with standard functionality.

Home		Save	New	Copy
Id	01-2015-002	Reg_No	01-2015-002	
Name	Saleem			
Program_ID	78 -- MBE	Batch	2015 -- 2015	
Father_Name	Akram			
Gender	M -- Male	NID		
DOB	01/01/2010			
Emergency Contact				
Email				
Permanent_City				
Permanent_Address				

### Search and Select View with Form

This UI view has three portions. The upper part contains a simple search form. When the user makes his search, the results are listed in the lower left part. In the example below, students are being searched.

After searching, the user clicks on a student. His details are displayed in the right portion, in an automatically constructed form. User can update data and press the Save button.

Apart from a reasonable runtime speed (response time), this offers two important types of efficiency:

1. It can be developed and deployed in hours.
2. It has been observed that it saves time of user. The user has powerful options on a single page.

The screenshot displays a web application interface. At the top, there is a search form with two input fields labeled 'Name' and 'Reg No', and two buttons: 'Search Now' and 'Add New'. Below the search form is a table listing search results. The table has two columns: 'Reg\_No' and 'Name'. The results are as follows:

Reg_No	Name
01-2014-001	Abdul Mubeen Malik
01-2014-002	Ali Mustafa Qureshi
01-2014-003	Amir Irshad
01-2015-001	Amjad
01-2015-002	Saleem
01-2015-003	Abdul Ali Asad
01-2015-004	Abdul Mateen Malik
CIIT/FA01-BCE-006/WAH	Arif Imran
CIIT/FA01-BCE-007/WAH	Arsalan Haider
CIIT/FA01-BCE-008/WAH	Arsalan Shafique
CIIT/FA01-BCE-009/WAH	Asad Shabbir
CIIT/FA01-BCE-010/WAH	Asif Majeed
CIIT/FA01-BCE-011/WAH	Asif Mirza

To the right of the table is a detailed form for a selected student. The form has a header with 'Home', 'Save', 'New', and 'Copy' buttons. The form fields are as follows:

Id	01-2015-001	Reg_No	01-2015-001
Name	Amjad		
Program_ID	64	Batch	2015 -- 2015
Father_Name	Sajjad		
Gender	M -- Male	NID	123
DOB	01/01/2010		
Emergency Contact	03123		
Email	abc@xyz		
Permanent_City	Islamabad		
Permanent_Address	Islamabad		

dfSearchSelect dfRichForm

## Quick Search View

This is a simple UI view in which the upper portion contains a simple search form. The user enters his search criteria and clicks on 'Search Now' button. The results are displayed in the lower portion.

Home Offered Courses All Sessions											
Session		ALL	Title		pakistan		Teacher		Dept		
Search Now		★									
ID	Dept	Program	Semester	Section	Course_No	Title	Credits	Teacher	Session	Session_ID	Detail
27350	MS	BBA	2		HUM111	Pakistan Studies	3	NA	FA01	1	detail
27676	MS	BBA	2		HUM111	Pakistan Studies	3	NA	FA01	1	detail
27611	CS	BCS	2		HUM111	Pakistan Studies	3	NA	FA01	1	detail
27329	CS	BCS	2		HUM111	Pakistan Studies	3	NA	FA01	1	detail
28396	CS	BCS	2	X	HUM111	Pakistan Studies	3	NA	FA01	1	detail
28413	EE	BEC	2	X	HUM111	Pakistan Studies	3	Mr. Masood -ur-Rehman	FA01	1	detail
28430	EE	BET	2	X	HUM111	Pakistan Studies	3	NA	FA01	1	detail
27355	CS	BTN	2		HUM111	Pakistan Studies	3	NA	FA01	1	detail
27854	CS	BTN	2		HUM111	Pakistan Studies	3	NA	FA01	1	detail
27867	CS	BTN	2		HUM111	Pakistan Studies	3	NA	FA01	1	detail
21	EE	BCE	2		PS1001	Pakistan Studies	2	Shahzad Zafar Chaudhry	SP02	2	detail

## Hierarchy View

This UI view provides a powerful way to navigate through a large data set. In the example below, the user can see the attendance of any student of a university. The left most frame gives a list of classes. When the user clicks on the desired class, the students of that class are listed in the next frame. Then user clicks on the desired student. His attendance is displayed in the right frame.

The developer needs no programming to provide this view to his users.

Course_Title	present	Absent	Out_of	Percentage
Enterprise Systems Development	12	1	13	92.31
Computer Communication and Networks	15	0	15	100
Automata Theory	14	1	15	93.33
Financial Accounting	10	2	12	83.33
Network Security	14	1	15	93.33

## Parent-Child View

In this UI view, the data of a parent entity is shown in the upper portion. The related data of the child view is listed in the lower portion. The lower part contains various tabs for various data. User clicks a button to see the desired information.

This view is especially useful when user wants to focus on some object, and explores it in detail. The example below shows a purchase order and its items. This is constructed automatically with Data Furnishing Tools.

id	6	PO_No	P009				
PO_Date	1970-01-01	Demand_No	1				
Supplier	2	Name	Telerisen Pvt. Ltd.				
Address	H No 295, Street No 15, I-10/1		Phone	0514101221			
City	Islamabad						

Home 6 rows

id	Item Code	Name	Demanded_Quantity	parent_id	PO_Quantity	Unit Price	Description
1	FUR0001	Chair	20	6	<input type="text" value="20"/>	<input type="text" value="2000.00"/>	Chair
2	CMP_001x	Computer 1	15	6	<input type="text"/>	<input type="text"/>	Computer 1
6	FUR0010	Table 1	12	6	<input type="text" value="12"/>	<input type="text" value="8000.00"/>	Table 1
7	FUR0011	Table 2	9	6	<input type="text" value="9"/>	<input type="text" value="9000.00"/>	Table 2
9	FUR0002	Chair 2	13	6	<input type="text" value="1"/>	<input type="text" value="3000.00"/>	Chair 2
10	CMP_001x	Computer 1		6	<input type="text" value="4"/>	<input type="text"/>	Computer 1

## Entry Table View

This UI view is similar to Quick Search view, but in the lower part, user can enter / update the contents. Some fields are display-only, while others can be updated.

[Home](#) (Specify your search criteria below and click on 'Search Now') **Plan Courses Update**

Program  Title\_  To\_Session

[Home](#) 888 rows [Select program, type 2099 in To\\_Session and search. You can search by title. Check courses and correct them.](#)

ID	Program	Course_No_	Title	Semester	is Current	is Elective	Offer	Prerequisites	Credits_	Lab_Hours_
3669	BBA	MGT100	Introduction to Business	<input type="text" value="1"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0
3670	BBA	HUM110	Islamic Studies	<input type="text" value="1"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0
3671	BBA	ECO100	Micro Economics	<input type="text" value="1"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0
4058	BBA	MTH108	Business Mathematics I	<input type="text" value="1"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0
4149	BBA	HUM100	English Comprehension and Composition	<input type="text" value="1"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0
4154	BBA	CSC101	Introduction to Computing	<input type="text" value="1"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	1
4151	BBA	MGT101	Introduction to Management	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0
4156	BBA	MTH109	Business Mathematics II	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	MTH108	3	0
4059	BBA	ECO101	Macro Economics	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0
3654	BBA	HUM111	Pakistan Studies	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0
3655	BBA	HUM102	Report Writing Skills	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	HUM100	3	0
4570	BBA	MGT130	Accounting I	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		3	0

dfEntryTable

## Main Panel

The Main panel displayed just after the login contains a dynamic menu. The options available in the menu are based on the user group, and also depend upon the mapping of data groups to user groups. DataFurnish provides a convenient way to allow specific user groups to access some options or stop them to see some items.

When there are hundreds of options in the menu, user can search for desired menu item. User can assign additional keywords to menu items, so that the search is easy the next time. Communication system displays few messages for the user if present, and there is an option to see all messages and comment on them.

<b>Accounts</b>	Logged in: najmalikram@yahoo.com
<b>Admission</b>	<input type="text"/> <input type="button" value="Search"/> <input type="button" value="Sign Out"/>
<b>Alumni_Admin</b>	
<b>Attendance</b>	New messages for you:
<b>Campus Admin</b>	Assalamualaikum Sir I am facing a problem while login to Microsoft team account for online classes. Solve my issue as soon as possible. (CIIT/
<b>Clearance</b>	<a href="#">See All</a>
<b>Communication</b>	
<b>Coordination</b>	
<b>DBA</b>	
<b>DCO</b>	
Answer Books	
Record For Sesionals	
Combine Courses	
Credits Remaining	
Dropped Courses	
Edit Student Sections	

## Other Features

Some generally required features are available with DataFurnish.

**Email:**

A number of different email types can be configured in metadata, causing system to send appropriate emails to users. For a reporting email, a query will be configured, specifying data group and schedule. The report will be emailed to all the users mapped to the data group.

For problem-reporting emails, a query is written such that it should return 0 rows under normal conditions, so the non-zero rows correspond to an error situation. The system will send automatic email to concerned users in case of non-zero rows.

Another email is where the message is a fixed text, and a query is written, that returns the (email addresses of) target users, for example, students who have deposited fee but did not register courses.

**Domain knowledge management:**

This feature allows users to enter text regarding various topics of the domain knowledge regarding the business of the organization. This may include procedures, rules, guidelines etc. Users ask questions, the designated people reply, and reply may be transferred to knowledge base.

**Application**

DataFurnish is particularly useful for systems with large number of database tables, or large number of possible queries. This is frequently the case in information systems.

One good application scenario will be to use DataFurnish side by side with a custom built information system. The custom information system would have few but finished features. DataFurnish would operate on the same database, but have independent user interface. Several information management functions would be accomplished through DataFurnish, which would support the overall system.

In this arrangement, DataFurnish would not interfere with the custom (or main) system, which could have been developed in a different programming language and using a different platform. Separate user(s) at database level will be created for DataFurnish, where the access rights can be limited if required.

**Development and Testing**

DataFurnish Tools have been implemented in PHP. It was deployed in a university environment using Microsoft SQL Server database, where it is running successfully for more than 6 years. More than 400 data objects were created for various requirements, a large number being Quick Search UI views. In rare cases, custom code was written to meet the specific requirements.

More than 6000 users have performed more than 7 million operations (e.g. viewing a report). This means, if we add a new feature using a DataFurnish UI view, it will be running through a well tested code.

## **6. Limitations of Data Furnishing Tools**

Following are the limitations of the tools

1. DataFurnish is suitable only for application that are built on a database having many tables and many possible queries.
2. DataFurnish has been implemented as a Web application. Mobile version is not yet implemented.
3. Some time, a system has a single main screen, that is heavily used and has special requirements. Such screen should be developed using customized code. However, other functionalities can be implemented in DataFurnish.
4. Features that require custom finishing or other specific things, should be developed separately.

## **7. Advantages Offered by Data Furnishing Tools**

DataFurnish tools make possible the transition from manual system to a computerized one through a speedy and convenient way.

With DataFurnish tools, the method of developing a software system is different. The tools have capability of “generating” major part of an information system. This unique power enables us to complete our computerization project far earlier than the others do.

Many software development tools and methodologies require us to freeze our specifications at a very early stage. Since DataFurnish tools can generate forms, tables and other elements automatically, modification in all these elements is also handled automatically. This enables the designer to make adjustments and improvements at later stages of development.

If the people are not very clear about the software requirements, or they want to give more and more ideas during development, then DataFurnish tools are particularly advantageous. With DataFurnish tools, it is possible to see the First prototype within weeks and to add more and more features as the development progresses.

Except a small portion, the system will be constructed without writing code. In other words, the system will be “defined” rather than “written”. This eliminates the chances of many well known complications that are related to writing code and maintaining it. The software industry frequently suffers from such problems, which include unstructured coding, ever growing bugs, tedious testing, dependency on programmers and many more. The code that is part of DataFurnish tools is highly generic, and is tested.

When the software system is expanded horizontally and more data objects are implemented, DataFurnish Tools give rich features to these objects automatically. So we get a large number of screens with standard features. These features enable the user to operate with power. Without DataFurnish Tools, the developer would not afford to give such features to large number of screens, and the software will suffer from limitations.