

Using XML to Manage the Abstraction Layer

Extensible Markup Language (XML) files and technology allow the customization of input/output into an abstract web-enabled database with amazing flexibility. This technology allows one database to accommodate disparate data and reporting requirements across a theoretically unlimited array of regulatory data collection and harvesting issues without the writing of a single line of code. And it allows the updating and editing of the individual data collection within minutes.

One of our oldest and most favorite clients is a business complex that includes the largest medical billing company in New Jersey. Over the past 20 years, Paladin has consulted and built systems which focus on that field. And during our association with this particular client, their medical billing revenues have increased twenty fold. That company had the savvy and foresight to be in on the recent boom in construction and management of these Ambulatory Surgery Centers (ASC), and nationally, they are known players in the field. Unfortunately, the boom in construction of ACSs across the country is not in a period of over expansion, and my client has turned to the business of rescue and turnaround of less profitable centers in other states.

Paladin's contribution to that effort is to provide the systems design and support for the program which does all of the billing of patients and their insurance companies, as well as provide the reports for management, and reporting to the various state regulatory agencies. With the expansion of ASC management within the firm, we have now become responsible for different and changing reporting requirements in various different states. These state agencies can and do change their formats and requirements as often as they please, and the management companies must comply with an ever changing and expanding array of data capturing and reporting requirements which are multiplied by the number of states in which we are practicing. In the state of Florida alone, over the past six months there were 4 changes of requirements and data capture/formatting with which we had to contend.

Obviously, the software demands for all of these changes was a challenge – particularly in light of the fact that each state had different demands for patient data.

Our solution was two layers of data abstraction. First, we designed an XML file for each state (an idea which could conceivably be extended to each center). We knew that the process of patient billing required a known set of fields to process a claim. And we knew that each state had different information that it required for the reports on the demographics of the patient. Second, the billing process was separate from the patient intake process, and a fair piece of the information required for the state reports had little to do with the billing process. On the other hand, some of the information at intake could be useful to capture in the billing program. So we

Written by Chris Kemp

Tuesday, 06 July 2010 21:46 - Last Updated Tuesday, 06 July 2010 23:58

revised our database tables to include a set of fields required for billing, and others, required by the states. A fragment of one of those XML files follows:

```
<?xml version="1.0" encoding="UTF-8"?> <FACILITY_RECORD> <PATIENT_RECORD>
<field type="text" label="First Name" name="FirstName" billingname="Firstnam"></field> <field
type="text" label="Last Name" name="LastName" billingname="Lastname"></field> <field
type="text" label="Street" name="Street" billingname="Street"></field> <field type="text"
label="City" name="City" billingname="City"></field> <field type="text" label="State"
name="State" billingname="State"></field> <field type="text" label="Zip" name="Zip"
billingname="Zip" AHCA="PATIENT_ZIP"></field> <field type="text" label="Country"
name="Country" AHCA="PATIENT_COUNTRY"></field> <field type="text" label="Phone"
name="Phone" billingname="phone1"></field> <field type="text" label="DOB" name="DOB"
billingname="DOB" AHCA="PATIENT_BIRTHDATE"></field> <!-- form is yyyy-mm-dd -->
<field type="text" label="SS" name="SS" billingname="SS" AHCA="PATIENT_SSN"></field>
<field type="text" label="Employer" name="Employer" billingname="Employer"></field> <field
type="text" label="Marital Status" name="MaritalStatus" billingname="Marital Status"></field>
<field type="select" label="Sex" name="Sex" billingname="Sex" AHCA="PATIENT_SEX">
<option value="M" selected="off">Male</option> <option value="F"
selected="off">Female</option> <option value="U" selected="OFF">Unknown</option>
</field> <field type="select" label="Race" name="Race" AHCA="PATIENT_RACE">
<option value="1" selected="off">American Indian or Alaskan Native</option> <option
value="2" selected="off">Asian</option> <option value="3" selected="OFF">Black or
African American</option> <option value="4" selected="off">Native Hawaiian or Other
Pacific Islander</option> <option value="5" selected="off">White</option> <option
value="6" selected="OFF">Other</option> <option value="7"
selected="OFF">Unknown (for use if the patient refuses or fails to disclose)</option>
</field> <field type="select" label="Ethnicity" name="Ethnicity"
AHCA="PATIENT_ETHNICITY"> <option value="E1" selected="off">Hispanic or
Latino</option> <option value="E2" selected="off">Non-Hispanic or Latino</option> <option
value="E7" selected="OFF">Unknown</option> </field> </PATIENT_RECORD>
<APPOINTMENT_RECORD> etc..... <APPOINTMENT_RECORD>
</FACILITY_RECORD>
```

In the XML file, fields are designated for the individual center, together with the menu of items they are allowed to enter. These fields can be related either to the patient, or to the appointment. Take, for instance the “Race” element (highlighted in red). Florida demands that Race be reported to it using a single digit code (1–7), but other states will require entries like “White” “Black” “Indian”, etc. The “Sex” field is another example: Florida requires a “M”, “F” or a “U” (unknown) entry. Other states require either an “M” or an “F”, or “Male” or “Female”.

The XML technique allows each state to have its own unique input form and validation scheme. A new field in the file triggers a new field to be created an auxiliary table in the database, which will reference, the center ID, field name, length, value and type of data expected. That field information will be accessed by the javascript in creating and validating and binding the data

Using XML to Manage the Abstraction Layer - Paladin Consultants, LLC

Written by Chris Kemp

Tuesday, 06 July 2010 21:46 - Last Updated Tuesday, 06 July 2010 23:58

from the input form, and mapping and binding the data to the particular field in the database. The sub elements in the XML file can refer to the patient table, or the appointment table, or, conceivably, any table in the data model.

Fields which are NOT unique to an individual center are included to allow data input taken at the center to be sent via a harvesting XML file to the billing system for inclusion therein. This eliminates duplication of effort issues.

When a new center is added to the ranks, a similar XML file is cloned and edited to accommodate the idiosyncrasies of reporting requirements for their local jurisdiction, and the uploaded to a common site. The new XML file is added to the protocol of the new center, and no further changes in coding is required. The whole process could take about 15 minutes. If a state or local government adjusts the information it requires, or its format, the XML file for that center is edited, and then uploaded to the hosting site, where it is processed immediately – again a 10–15 minute job resulting in not a single programming change.