

WYSIWYG Development of Data Driven Web Applications

Avadhoot Kalyankar (ak77@buffalo.edu)

Overview:

An emerging trend in Social Networking sites and Web portals is the opening up of their APIs to external application developers. Additionally, there is an increasing need to leverage web pages as computing platforms that can enable users to build their own applications. However, existing technologies are targeted towards developers with programming expertise and database knowledge. They are not accessible to a large class of users who do not have a programming database background. To address this need, the AppForge system has been developed by the authors, which provides a WYSIWYG application development platform. AppForge system allows users to create sophisticated stateful applications and complex views over multiple entities. Users have to specify the components graphically and the corresponding database schema and application logic will be automatically generated on the fly. The User at no point has to worry about the underlying schema or application logic. Also the WYSIWYG interface gives instantaneous feedback on what users have created and allows them to run, test and continuously refine their applications.

AppForge consists of

- Application Creation system: It creates and updates the application model and stores it onto file system. The application state is stored in the relational database.
 - Page View Creation Module: Creates and updates web pages.
 - Automatic Schema Generation Module: Generates database schema
- Application Runtime system: Loads the application model and interprets it.

AppForge GUI consists of the following abstractions:

- Application: Users can create and manage multiple applications.
- Role: Users are divided into different roles/categories.
- Page: Different web pages.
- Form: User input component to enter new data.
- View: To view the data
- Container: Entities in an application.

The application model of AppForge is based on the E-R model and Non-Relational Algebra model. The first normal form is relaxed for the model. Further, application model consists of

- Database Model: Specifies the application state including schema and constraints.
- Page Model: Specifies the content, structure and presentation of web pages.

Page View Creation Module is in charge of displaying the schema without exposing neither the complexity of an E-R graph or the details of NRA operators such as join or nest. This is done by creating a Schema Navigation menu. The navigation menu is a tree structured menu. The algorithm is applied for each entity and the tree is generated.

Automatic Schema Generation is in charge of generating database schema on user's actions. Database schema is generated for new form or when adding columns to a form.

AppForge has been used to create prototype versions of a variety of applications such as an event planning system, a recruiting system, an item trading system and an online course management system. A small and preliminary user study to identify and fix some of the usability aspects of AppForge has been conducted.

The user study was conducted on pairs of three pairs. The pairs were:

- Group 1: Two researchers with advanced degrees and involved in databases.
- Group 2: Two researchers with advanced degrees but no experience.
- Group 3: Experienced users but with no formal training.

They were given a test application and given an hour to complete it. The entire session was videotaped. Group 1 found the application very easy to use. Group 3 was the most challenged and were confused by the different levels of abstraction provided.

The feedback from the test was the need to improve the database terminology and user interface.

Detailed Comments:

1. The application would serve a good purpose if used to create prototypes. This would enable the developers to visualize the application and show it to the intended users.
2. One of the points observed by the audience was that it is limited to database applications which require some kind of user input data and form view. There is not much application logic creation involved. Thus the application is limited.
3. Also, currently the AppForge system is not yet open to the public for use. Their tests are not too successfully with the category of users without programming expertise. The changes made after the user feedback are not shown.
4. The paper is very clear about the AppForge system and gives a very good idea about the application and its intended use. The snapshots walk through the application.
5. The technical paper overall explains the concept of schema generation and view creation succinctly. Also the underlying theorems and logic has also been explained.
6. The weakness about the paper as observed by the audience is that the application might end up being used for prototyping. The paper should have been clearer about the changes after the tests.