

# BioGenex Proposal

Author: Michael Benson (mbenson@earthlink.net)  
History: 07/19/2003 – first draft

## 1 Background

BioGenex has a very wide range of products and technologies. This makes it very difficult to keep the web site up to date. The solution is a database driven system in which BioGenex can update products, technologies, and news events, and then have the changes reflected on the web site automatically.

This document describes the proposed technical details for implementing such a system.

## 2 Use-Case Scenario

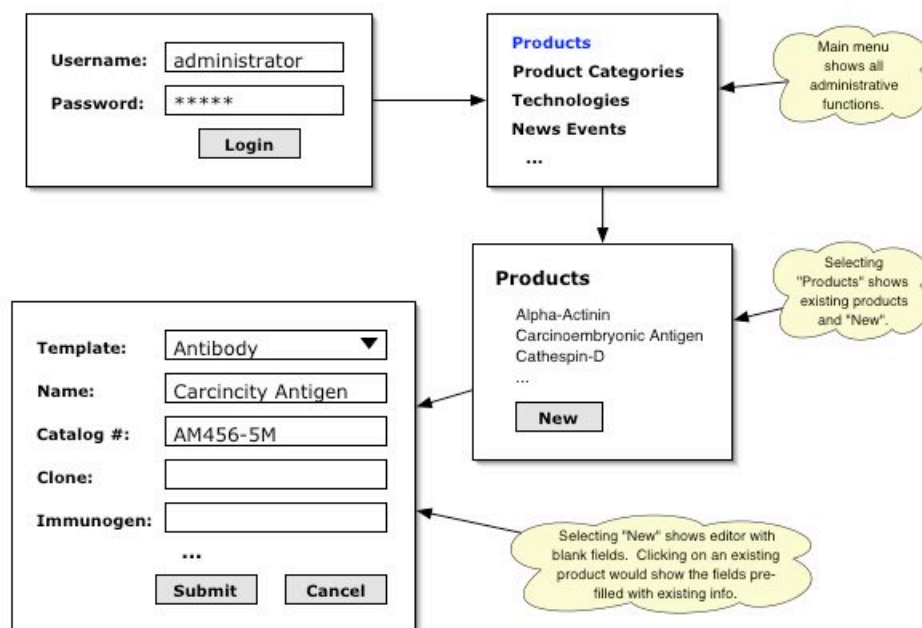
This section describes a use-case scenario for updating products and news items in the database, and viewing the new pages.

### 2.1 Updating the Database

BioGenex has a new product, the “Carcinicity Antigen”. A BioGenex administrator logs into the database maintenance system using any web browser and a username and password. On the main screen, the administrator chooses “Products”, then “New Product” and then chooses the appropriate template for Antibodies. The Antibody editor page then appears with fields for product name, product number, clone, immunogen, source, specificity, protocol, image name, etc. The administrator fills in the fields and clicks the “Submit” button. The product is then added to the database.

If BioGenex decides they need a more secure system, they may want the button to say “Submit for Approval”. This would add the product to a temporary area of the database and send an email to the lead administrator who approves all changes. That administrator would then review the changes and submit the product to the live database. This system would require each user to have an account with a password (as opposed to just a single login) and the lead administrator would need to have access to an approval administration page. Plus, the database would be considerably more complex because it would have to keep track of pending changes, and editing pages would have to be session based in order to keep track of who is making what changes.

The following diagram shows the basic process of adding a new product:



## 2.2 Viewing the New Page

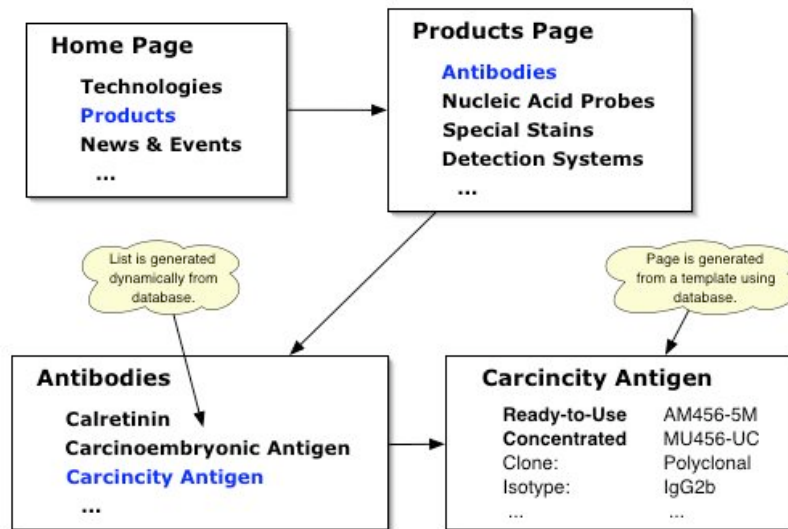
Once the product has been added to the database, it can be viewed on the live web site in three ways: (1) navigate to it through the Product pages, (2) navigate to it through the Technologies pages, and (3) do a search for it.

### 2.2.1 Navigating Through the Products

To get to the newly added product, you can navigate through the appropriate static pages to get to the “Antibodies” page (Home -> Products -> Antibodies). This page will have a dynamically generated list of all products in that category. This list could be presented visually in any number of ways. The items in the list would not be hard links to static pages. Instead, they would be links to a script (CGI/Perl, PHP, or whatever) containing the parameters necessary to get the item from the database. The script would then retrieve the item from the database and generate the display page dynamically from the appropriate template.

If the list of products in any one category is too long, you could subdivide them into subcategories. These subcategories could also be used to generate pages dynamically, or they could just be static pages like any other top level category.

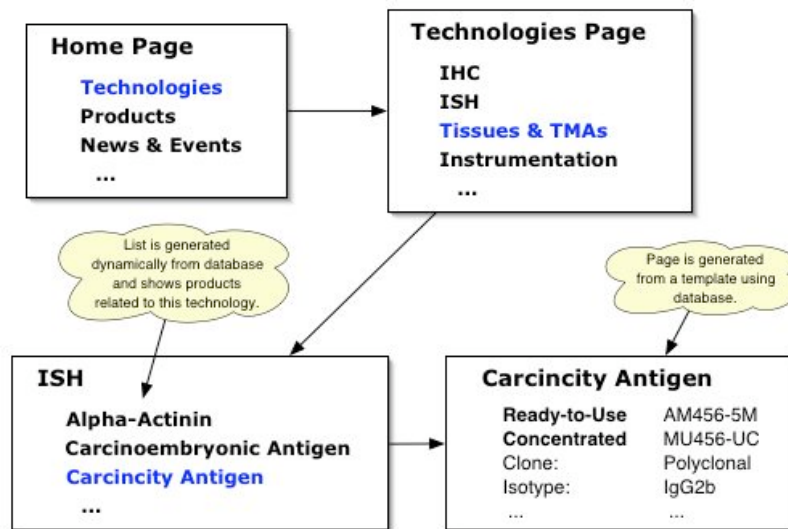
The following diagram shows the basic process of navigating through the products:



### 2.2.2 Navigating Through the Technologies

When products are added to the database, the administrator would specify which technologies the product is applicable to. So, when you navigate through the static Technologies pages, there would also be dynamically generated lists of products which apply to the specific technology. This would work pretty much the same way as the Products pages. However, it is possible for a product to appear on more than one Technology page, and it is also possible that not all products will appear in the Technologies pages.

The following diagram shows the basic process of navigating through the technologies:

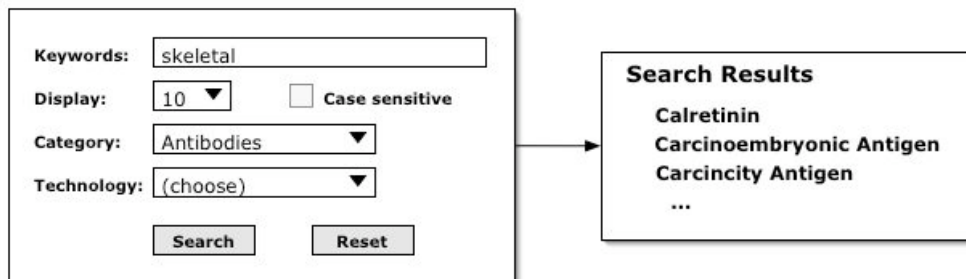


**Note:** Since there are so many products, and you may not want them all to be accessible through the technology pages, the Product Editor could have a way of specifying whether or not you want the product to appear in the list on the relevant Technologies pages.

### 2.2.3 Searching for the Product

A search function will be available from a link or button at the bottom of every page. The search page will allow the user to search by keywords, categories (such as Antibodies, Detection Systems, etc.), and technologies (such as IHC, Instrumentation, etc.). The categories and technologies will be drop-down lists populated from the database. If the user selects the appropriate search parameters, the newly added product will appear in the search results. The search results are presented visually in the same way as described above, i.e. the links contain the necessary parameters to generate the page dynamically from the database.

The following diagram shows the basic process for searching for a product:



## 3 Administration

The administration pages would be accessible from any web browser with a username and password. Depending on the level of security you want, you may also want to have separate user accounts. The administration pages would be used for adding and editing products, product categories, technologies, news events, and possibly other things like FAQ's, contact info, job openings, etc.

In addition, if BioGenex decides to go with the administrative approval system described above, there would be pages for managing user accounts, reviewing pending changes, and doing the final approval.

### 3.1 Editors

When the user logs into the administration system, the main admin page appears. This would simply contain choices for the various available actions (edit products, edit product categories, edit technologies, etc.). Selecting one of these items will bring up the appropriate editor.

#### 3.1.1 Product Editor

When the Product Editor is selected, a list of all existing products will appear alphabetically (possibly with paging since this list could be very long). The user can select a product to edit, or click the "New Product" button. This will bring up the actual editor with the various fields to be filled in. In the case of an existing product, the fields will be pre-filled and the user can edit them. The actual fields in the editor will depend on which template the product is using, however that can be generated dynamically from the database, so only one editor is needed. (See section on Database for more details.)

When all changes are complete, the user would click "Submit" (or "Submit for Approval") and the database will be updated.

### **3.1.2 Product Category Editor**

The Product Category Editor may or may not be necessary depending on how often the categories change. If the categories are fairly constant, then you could just update the database manually on the rare occasions that you add a new category. But, if categories are changing frequently, then you may wish to include this editor.

The Product Category Editor would show a list of all existing categories plus a “New Category” button. Selecting a category (or new category) would bring up the actual editor with the various fields to be filled in. In the case of categories, the fields are fixed (name, description, URL, parent category, etc.), so no templates are needed.

### **3.1.3 Technologies Editor**

The Technologies Editor will be very much like the Product Category Editor and may or may not be necessary depending on how often the technologies change.

### **3.1.4 News & Events Editor**

On the current BioGenex web site, there are basically two types of News & Event items: Press Releases, and “Other”. In the Press Release category, the items are presented in a standard format, so this could be done with a template that is filled in from the database. In the “Other” category are things like corporate newsletters (“BioLink”), trade shows schedules, workshops, etc. These are all basically pointing to static web pages.

The News & Events Editor would allow you to add or edit items in both categories. In the case of Press Releases, the editor would have fields for date, headline, copy (the actual text content), etc. For the “Other” items, the editor would simply have the link (or URL) to the corresponding static page.

### **3.1.5 Other Editors**

Depending on what other items you wish to maintain dynamically, there might be other editors for things like FAQ’s, contact info, job openings, etc. These editors would all be very simple and straightforward and would not require templates or categories.

## **3.2 User Administration**

If BioGenex decides to go with the administrative approval system described above, special administration pages would be needed for maintaining user accounts, reviewing pending changes, and doing the final approval. These pages would only be accessible to the lead administrator.

### **3.2.1 User Accounts**

The User Accounts admin page would show a list of users. You can select an existing user or click “New User” to bring up the user editor. This would allow you to set the username and password, real name, email address, and level of access (i.e., a regular user or an administrator). It’s up to the lead administrator to maintain this information and make sure the appropriate people have access to the system.

### **3.2.2 Reviewing Pending Changes**

The pending changes would be displayed as a list showing the user and date & time (“session”) of the changes. The administrator would select a session from this list to review the changes. This would allow the administrator to browse through the changes using the same editors as described above, but the changes would be highlighted (probably by color).

When the administrator is ready to approve or disapprove the changes, he/she would click a link to go to the approval page. This would have buttons for “Approve” and “Disapprove”, plus a text area for entering a comment to send to the original person who made the changes (such as “Your changes are being disapproved because we’re discontinuing the product.”).

## 4 Database

The database contains all of the data necessary to drive the web site and the editors. The complexity of the database will depend partially on the features BioGenex decides to include in the system. The feature that will add the most to the complexity is the approval process. This will require tables that work in parallel to the live database plus extra fields for user sessions, and tables for user accounts.

This section describes the various tables that will be needed and how they interact with each other in the system.

### 4.1 Tables

There are at least 7 tables needed for the basic functionality of the system, and possibly more if BioGenex decides to go with the administrative approval system.

#### 4.1.1 Categories

The *Categories* table contains all available product categories such as Antibodies, Nucleic Acid Probes, Special Stains, Detection Systems, etc. Each record in the table would contain an ID and name for the category. The ID would be cross-referenced with the *Products* table.

#### 4.1.2 Templates

The *Templates* table contains all available product page templates. Each record contains an ID, a category ID, and name for the template. The ID would be cross-referenced with the *Template\_Fields* table. Normally, a template will correspond to a certain category, but it is possible for one category to have multiple templates, so a separate table is used for templates.

#### 4.1.3 Template\_Fields

The *Template\_Fields* table contains the descriptions of all fields in all templates. A field might be something like clone, isotype, description, etc. If it is determined that some fields are common to all templates (such as name, catalog #, etc.), these can be included in the *Templates* table.

#### 4.1.4 Technologies

The *Technologies* table contains all available technologies. Each record contains an ID and a name. The ID would be cross-referenced with the *Products* table.

#### 4.1.5 Products

The *Products* table contains all available products. Each record would contain an ID, product name, template ID, and possibly other information. Most of the specific information about a product is actually contained in the *Product\_Fields* table described below. This is done because the number of fields describing a product is variable – it depends on the template it uses. So, the actual data must be contained in a separate table.

#### 4.1.6 Product\_Fields

The *Product\_Fields* table contains all of the fields describing all of the products. Each record would contain an ID, a product ID, a template field ID, and the actual data for that field. This table

is cross-referenced with both the *Products* and *Template\_Fields* tables to extract all the relevant fields for the product. This information is then used to fill in the templates when generating the web page.

#### 4.1.7 Product\_Technologies

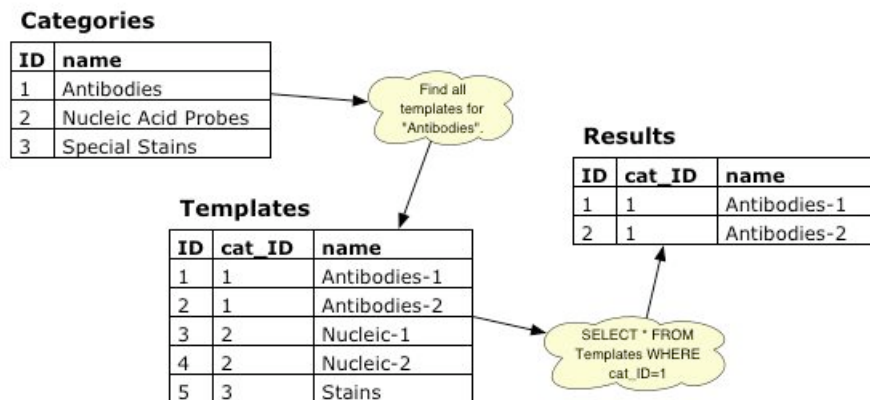
The *Product\_Technologies* table is used for relating the products with all the relevant technologies. Each record contains a technology ID and a product ID. If it is determined that a product can only relate to one technology, then the technology ID could be included in the *Products* table and the *Product\_Technologies* table could be eliminated.

## 4.2 Table Interactions

This section describes how the various tables interact with each other to extract the information needed for the web site.

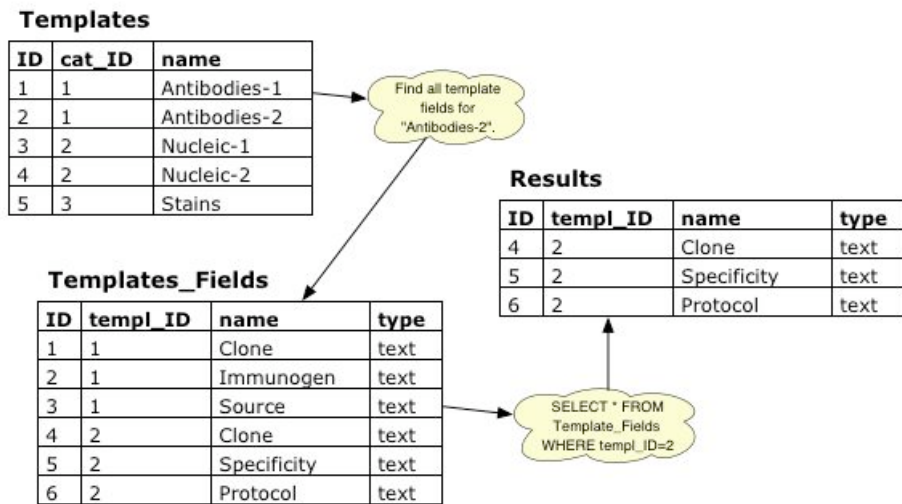
### 4.2.1 Finding the templates for a category

A category can have more than one template, so to find all the templates relevant to a given category, the category ID would be cross-referenced with the *Templates* table like this:



### 4.2.2 Finding the fields in a template

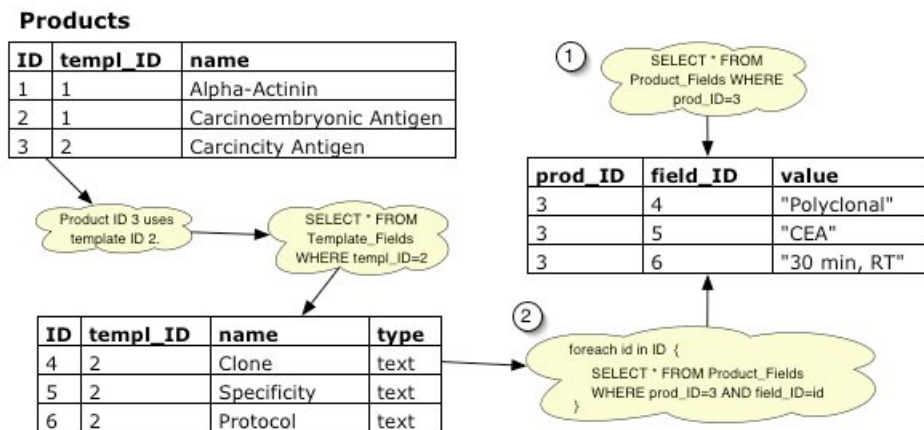
Once the appropriate template has been found, we might want to get the descriptions of the fields in that template. This would be done by cross-referencing the template ID with the *Template\_Fields* table like this:



Notice the *Template\_Fields* table has a column for *type*. This may be necessary if we decide the data for a given field can be something other than text. However, since the meaning of the field is dependent on the context of the actual web page, this probably won't be necessary.

### 4.2.3 Finding the information for a product

Now that we have found the fields for a given template, we can find all of the information relevant to a product and fill in the template to generate the web page. Since the *Products* table contains a template ID, we can get the relevant fields as shown above. Then, we would query the *Product\_Fields* table to get the actual values like this:



In this diagram, notice we find that product ID 3 ("Carcincity Antigen") uses template ID 2 ("Antibodies-2"). We can then get the template fields for that template and loop through them one by one to get each field value for the specific product (yellow cloud #2), or we can just do a query to retrieve all of the values at once (yellow cloud #1).

## 5 Web Page Generation

The ultimate goal of this project is to generate web pages dynamically from the database, so this section describes how this will be done.

### 5.1 The Templates

The page templates will consist of either a CGI/Perl script with embedded HTML, or an HTML file with embedded PHP scripts. There are probably many other ways this could be done, but these two are the most likely candidates.

The template will have all of the necessary page layout elements with scripts to fill in the variable information. The scripts will use parameters passed into the page such as product ID to determine what values to fill in. For example, a template might look something like this:

```
<?php
    $id = $_HTTP_GET_VARS["prod_id"];
    $clone = GetClone($id);
    $specificity = GetSpecificity($id);
    $protocol = GetProtocol($id);
?>

<html>
<head>
    <title><?php echo $prod_name; ?></title>
</head>

<body>

<h1><?php echo $prod_name; ?></h1>

<table>
    <tr>
        <td>Clone:</td>
        <td><?php echo $clone; ?></td>
    </tr>
    <tr>
        <td>Specificity:</td>
        <td><?php echo $specificity; ?></td>
    </tr>
    <tr>
        <td>Protocol:</td>
        <td><?php echo $protocol; ?></td>
    </tr>
</table>

</body>
</html>
```

Notice the embedded PHP script gets the appropriate field information from the product ID which was passed into the page via the HTTP request. Then, it calls functions which retrieve the values from the database. (Note: The actual code is unlikely to look like this.) This script runs on the server filling in the values and then sends the generated page to the web browser.

### 5.2 Getting to the Web Page

As discussed in section 2 (Use-Case Scenario), there are several possible ways to get to a dynamically generated page. But ultimately, they all involve using a URL with the appropriate parameters to retrieve the data from the database. So, for example, say the PHP script in the

previous section is named “template1.php”. To display the product Carcincity Antigen, the link on some page to that product would look like this:

```
<a href="template1.php?prod_id=3">Carcincity Antigen</a>
```

It is also very likely that the link itself would be dynamically generated on another page. For example, on the Antigen category page, the list of antigens would be generated from the database, so the template name and product ID would also come from the database.

## 6 Task Breakdown

This section discusses the major task breakdown for the project with approximate time estimates for each task.

### 6.1 Database Design and Testing

The database tables and operations can be designed and tested without any GUI front end. Scripts can be run on the server to create the database, initialize it with test data, and perform various operations like finding the fields that belong to a given template. The results can simply be displayed or printed and verified.

This process should take no more than 2 to 3 days unless BioGenex decides to go with the administrative approval model. This would probably add at least a few more days and possibly a week. Also, this does not include the actual database initialization. (See “Integration and Testing” below.)

### 6.2 GUI Design and Testing

There are three parts to the GUI design:

1. The static pages
2. The templates
3. The scripts

The first two parts will most likely be done by BGD1, so I’m not including that in my estimate. For the third part, I would be taking the templates and integrating in the necessary scripts to generate the dynamic content. Depending on the number of templates, this could take anywhere from 2 to 4 days.

There are also other pages which require dynamic content such as the category pages (the list of products in the category is generated dynamically), the technologies (the list of products related to the technology is generated dynamically), the search page, and possibly other pages like news, FAQ’s, etc. These pages would also require HTML templates (but only one for each page type) to which I would add the appropriate scripts. This might add something like a few days to a week to the project.

### 6.3 Administration Design and Testing

The system can actually go live without any administration pages by just initializing the database from a script. However, without the admin pages, any future maintenance would require someone to manually edit the database.

At a minimum, the Product Editor page will be needed. This page will probably take something like 2 or 3 days to design and test with several templates.

If BioGenex decides they need the other editors, the Category Editor and Technology Editor will take something like 3 to 5 days combined to design and test.

If BioGenex decides to go with the administrative approval model, this will require several admin pages (user account administration, user edit session viewer, approval page, plus modifications to the editors to view user's changes with highlighting). This could easily add something like a week.

## **6.4 Integration and Testing**

The final phase will be where all of the pages (both static and dynamic) are integrated and the database is initialized. Gathering up all of the data and creating a script to initialize the database will probably take something like a day or two of my time, but it will probably also take a significant amount of time for BioGenex to format this data in something like a spreadsheet. Then, testing the integrity of the database will also require some scripts to check out various table relationships. This will add another day.

## **6.5 Totals**

Depending on which parts BioGenex decides to implement, the time estimates would range as follows:

1. No category or technology editors, no admin approval: 11 to 18 days
2. All editors, but no admin approval: 14 to 23 days
3. All editors and full admin approval: 24 to 33 days