Don Gardener Legacy Database Project Summary
Prepared by: Richard Wallsgrove

This report is intended to summarize the work done in preparing the DGLegacy database, identify problems encountered during its creation, assess its current state, reflect on the lesson's learned during the project, and comment on the possibility of creating future Legacy databases.

Summary

When I began the project, the database consisted of a bibliography of Don's publications, hard copies of most of these publications, and a set of 50 CD's containing images from the span of his career. The first step was to place the bibliography in a more suitable form (I chose Excel for it's easily navigable interface), parse the entries, and devise a scheme to connect the publications to the images that Don noted were associated with each.

The next step was to turn Don's hand-written notes on each image into a digital index of the images. This was the single most time-consuming leg of the project, taking upward of two months to complete, not including the numerous revisions undertaken since. The difficulty in this task stemmed from several sources. Most simply, Don's notes were very difficult to read. Since the information in each note was very important (often indicating the identification of the host and/or pathogen depicted), I took great care to make sure that simple things such as spelling were correct. This involved using several reference sources to find a match to Don's sometimes cryptic naming scheme. This was especially difficult given my limited background in plant pathogens. Rob Anderson was always available to provide me with necessary help, but it simply was not efficient to consult him with every question that arose.

Another problem I encountered was identifying those images that were important to the project. Initially, I tried to create a complete index of all the images for the sake of being thorough. I soon realized that this was far too time consuming, and limited my efforts to images of scientific interest. Since there was no apparent organization to the images to start with, this meant individually selecting or deleting entries in the index until it had been acceptable culled of extraneous images.

The third step was digitally transcribing the abstract for each publication. While not difficult, this was time consuming. This was followed by scanning the hard copy of each publication to create a PDF file that allows the incorporation of full text references into the database. In truth, the most difficult part of this task was attempting to find the most efficient way to do this. It was recommended by Phillip Thomas that each publication be scanned at high resolution, then "dumbed down" for use with the database. The logic behind this plan is that the most difficult part of scanning documents is physically placing documents on the scanner one at a time, and that high resolution scans may be useful in the future (presumably for optical character recognition). However, I was provided a scanner with a document autofeeder that very quickly and efficiently scanned the documents to Adobe Acrobat, creating a searchable PDF file. Although these files are

large, their creation took a very small fraction of the time that would have been required to create high resolution scans. Additionally, the poor quality of many of the original documents makes OCR an unlikely possibility.

At this point in the project, I was given a database of host/pathogen pairs that Don created some time ago. My work with this database was limited to converting it to MS Access (from Paradox), and relating it to Don's publications. It turned out to be an invaluable reference tool for identifying species that were not clear from Don's notes on his images.

Time was also spent identifying useful information for incorporation from Don's miscellaneous files, and from his website.

Current State of the Database

The database currently resides in MS Access, as three separate, but related, data tables. The database can be interfaced easily using Access, but requires the user to have direct access to the database files. In order to share the database more widely, work needs to be done to place the Access database on the web, or to create an html interface. This is best done by someone with thorough knowledge of CGI or similar languages.

Lessons Learned

The most obvious lesson learned from the Don Gardener database should be that the most time intensive part of database creation is almost always data entry. Complete data tables with a sound structure can easily be turned into a complete database by a competent programmer.

It also became clear that it would be more efficient to build Legacy databases if one began with a more clear vision of the final product. In this case, that was impossible. However, in future projects I recommend that the investigator whose "Legacy" is being created have a more direct role in the creation of a database "vision". This allows the database creator to spend less time familiarizing him or herself with the available (and useful) data, and more time getting on with things.

It should also be noted that this database has been designed with Don's data in mind. It is not particularly scaleable to other data sets. That said, it is not impossible to envision that this database can be combined with others to created a legacy "master" database. However, one must find a way to relate data from very different backgrounds (ex. data from a plant pathologist and ornithologist). In my opinion, there is only one way to accomplish this; through and index of the scientists' published works. While it may seem redundant to "republish" these works, I believe strongly that the bulk of any scientists' legacy will consist of their published works. Other data can easily be added to a framework of publications. I have consciously kept Don's publication data separate from other data with this future goal in mind.

Don Gardner Legacy Database Project Notes

IMPORTANT: Remember that the files MUST reside in a folder called C:\DGLegacy\ for the database to work. If this is impossible, you must edit the data tables to reflect the actual location of the files. You can replace "C:\DGLegacy\" with the proper location (a good idea would be to change this to the letter of your CD drive – such as "D:\". You must do this to the table PICS, in the field PicLink and PicLocation, in the table PUBS, in the field FullTextLocation.

For more info on making the database work, see the README.TXT file.

The database resides:

- in C:\DGLegacy on the Dell in Henke 337
- on the D: drive from the same computer, now removed and placed on Dave Helweg's desk
- on several CD's

NOTES: The original images in PCD format are too big to place on a CD, they reside on 50 CD's in Henke 337, and on the C: drive from above. The uncompressed JPGS from those files are also too big, and live ONLY on the C and D drives named above.

Another directory, DGProject also lives on the C: and D: drive. This folder contains old, miscellaneous files collected during the database's creation. Nothing in here will be useful unless a dire emergency occurs.

A form was created to add and edit the data tables, but is not found in the primary interface. It is called "Edit Tables".

The database consists of 3 tables: PICS, PUBS, and PATHOGEN

PUBS

This table contains references to approximately 150 of Don's published works, and another 50 unpublished notes on various subjects.

EntryID: Each reference is identified by a unique 3 digit number (stored as text) from 001 to 203. This field is named "EntryID" in the data table. This identifier can easily be enlarged to accommodate more listings by adding zeros in front of each EntryID.

FullDescription: Each entry has a full title (that includes title, journal, author, journal etc…). This full title has been parsed into it's individual pieces to allow it to be combined with other databases.

Abstract: Where applicable, each entry also has its abstract entered into a memo field. Because of difficulties in storing text formatting in the memo field, the abstracts were also placed into MSWord documents, each titled with the EntryID of their associated publication. If it is deemed necessary, these files can be placed into the database as OLE

objects, and displayed in forms and reports. This is less efficient that using the memo field, but will preserve formatting (italics for latin names etc.). In the event that this route is chosen, the field **AbstractFile** was created to test the storage of Word documents in the database.

Absract?: To keep track of which entries have an associated word abstract document, thie yes/no field was created.

Biocontrol?: Yes/no field tracks which entries are related to biocontrol. This information is also in the keyword field, using the term "biocontrol". The success of each biocontrol project was judged qualitatively from the article text. Most entries reach no definitive conclusion, and "no conclusion" is placed in the "**Success?**" field ("Yes" and "No" are the other possibilities in this field.

Host: A text field stating the host genus or species covered by the entry. Common names are put in the keyword field.

Pathogen: A text field stating the pathogen genus or species covered by the entry. Common names are put in the **Keyword** field.

PDF? and **FullTextLocation:** For applicable publications, an Adobe Acrobat .pdf file was created from a 300dpi scan. Most of the pdf's are searchable for text within Acrobat, but the files were not created using OCR software. Rather, Acrobat has created "best guess" metadata text for each file, but displays the files as images. The existence and location of this file is tracked in these two fields. This is related to the Hard Copy? field, but no pdf was created for entries that consist only of an abstract (ex. conference notes). FullTextLocation can be easily created from the EntryID and path leading to the pdf files.

PicsAll: This field stores the relationship between the PUBS table and PICS table. Images associated with each entry are stored using their 5 digit (##_###) identifier, listed and separated using a comma. See Relationships section for more information.

AbstractLocation: Stores the location of the Word document containing abstracts. This field is used only to create a hyperlink, and is easily created from the EntryID and path leading to the abstracts.

For the moment, most of these fields are text, except for FullDescription (memo), Abstract (memo), FullTextLocation and AbstractLocation (hyperlink), and the yes/no fields.

PICS

This table contains information about approximately 2500 images from Don's CD collection of images. The naming scheme for the image files comes from their location on the CD's. He has 50 CD's labeled 1-50. Each CD contains approximately 100 images in Kodak PhotoCD format (.pcd). The images were named from their two digit CD

location (01 – 50) and number on the CD (approx 001 to 110), separated by an underscore. Example: the first image is named 01_001.jpg.

The pdc files were converted to jpg files and several copies were made: full size, uncompressed images (3072 x 2048 pixels, approx 3-4MB per image), full size compressed images (3072 x 2048 pixels, approx. 100 to 500 kb per image), reduced compressed images (approx 400 x 400 pixels, 20-30 kb per image), and thumbnail images (150 x 150 pixels, 3-4 kb per image). At the moment, only the smaller compressed images are used in the database, to allow them all to fit on a CD and keep the database efficient. The name of each image is the same for each file size. This can easily be changed by adding text such as "thumb" to the front or back of each file name. You can use imaging software such as Adobe acrobat to do this as a batch rename, or use simple file renaming software.

Many of Don's images were not scientific in nature (almost ½). These are not tracked in the database. Information about some of his personal images can be found in an excel file (see Excel Files for more information).

I never was able to solve all the issues related to dynamically generating the location of the files containing the images. Instead, I've had to rely on the brute force solution of placing the full path of the images in a field in the table. Should the images be moved, this is simple to change with a replace. It's not actually that difficult to assign the location based on a path that the user gives upon startup. The trouble I had was dealing with errors when an incorrect path is given.

The PICS field consists of the following fields:

CD and Pic#: These fields merely track the original location of the image. This information is also found in…

ImgID: This is the unique identifier for each image, ##_###. This identifier corresponds directly to the file name of each image.

ImgNotes: Don hand wrote notes for many of the images inside the CD cover's. These notes are in this field. Where possible, his rather cryptic notes have been expanded to include host and/or pathogen genus and species. This information was then placed in the appropriate "**Host**", "**PicHostGenus**", "**PicHostSpecies**" etc. fields. For some images, Don also wrote down the data they were taken, stored in the "**PicDate**" field.

PicPubRefs: This field tracks which entries from the "PUBS" table each image is associated with. This is done using the EntryID of associated publications, which are simply listed as text, separated by a comma.

PicLink: Created merely to provide a hypertext link to each image. This field can be created from the ImgID and path to images. "**PicLocation**" is the same field, but w/o a hyperlink.

PATHOGEN

This table comes almost directly from Don's PATHOGEN database of host/pathogen pairs. Three fields were added:

Pathogenshort and Hostshort: these fields were created to store the names of pathogens and hosts, but without RAAB references.

DGPubRefs: These are used to reference entries from PUBS which are related to the host/pathogen pair.

Also, the LEAF, STEM etc. fields were changed to text fields (from yes/no) for display directly on a form. This can easily be converted back to a yes/no field.

Relationships

Creating relationships between the table using Access was difficult, because every relationship is "many to many", and must exist between tables of vastly different sizes. Instead, I created relationships directly. PUBS and PICS were related using their primary keys (PUBS.EntryID and PICS.ImgID) in the fields PUBS.PicsAll and PICS.PicPubRefs. On the forms, these relationships can be recalled by creating a query using these fields (for example, to find all PICS related to pub entry 001, I've searched PicPubRefs for the text string "001" and returned all matches to a query to be used on the next form).

The relationships PUBS and PATHOGEN is created in the same way, using the field PATHOGEN.DGPubRefs. Relating PATHOGEN to PICS can be accomplished by searching for matching host/pathogen pairs in each table.

While I believe that referential integrity is not necessarily threatened by this system, it does have some downfalls. One that I can think of is the following: if one wishes to change the format of the EntryID field in PUBS, you must also change it in PICS.PicPubRefs. This can be accomplished using a "replace" for each entry ID. Not trivial (nor efficient), but nonetheless possible.

EXCEL FILES

I did much of the data entry for this database in MSExcel. These files still exist, and match the Access tables with matching name. If future changes are to be made to data, the excel files can be used, but will have to be re-imported to Access. It is VERY important that field names be kept the same if you wish to keep the Access form working.

Another way to edit the data has been included on the Startup form: the Add/Edit Images, Pubs, and Paths buttons allow access to the tables directly. It is important not to attempt to edit data from the other forms, they DO NOT access the tables directly, but rather secondary queries that are deleted upon return to the previous form.