User Manual for Hamster

Andrew U. Frank

February 9, 2011

TU Wien, Department of Geoinformation und Cartography,
Gusshausstrasse 27-29/E127.1
A-1040 Vienna, Austria
frank@geoinfo.tuwien.ac.at

GZ 4879, svn 3821

1 Introduction

Hamster deals with three issues of importance in text priented offices, like universitiy and research institutes:

- Long term centralized storage: The management of electronic documents—produced by ourselves or received from others—and the long-term storage of them for a group of related workers.
- Centralized storage of paper documents.
- Management of versions of documents: Electronic documents tend to evolve through a multitude of versions, sometimes in linear succession, sometimes in parallel variants.

These issues are usually treated unsystematically and separatedly. Hamster combines teses in one tool.

2 General Solution

The central collection and storage of paper documents is traditionally achieved in an archive. Archives can be organized by persons (e.g., in a medical practice), by properties (e.g., a registry of deeds) or by "business cases" (e.g., in an engineering firm). For a university institute few efforts are repeated and the variability of different tasks lead us years ago to an organization by GZ (German for "Geschäftszahl", English term would be docket or reference number).

New GZ are opened for any task, which takes a substantial amount of time (e.g., more than half a day) and are chronologically numbered. The GZ and a descriptive text are entered in a database and all paper documents filed under this GZ number *centrally* and *permanently*. Documents are retrieved by text search in the descriptions for the GZ and then the paper documents are found in the file cabinets.

This system works for most limited tasks but is not applied for continuous tasks like documentation of accounting and similar permanent tasks.

The increasing amount of documents produced and communicated electronically, forces an extension to electronic documents, dealing with versions of documents and reducing unnecessary duplications of storage of minimally changed documents posed an additional requirement.

2.1 Principles

- Centralized organization of files
- For all team members
- Jointly for paper and electronic documents
- Flexible
- GZ based (filenames are for information only)
- Search in a free text title database

2.2 Elements of Solution

A relational database (MySQL) is used to produce the GZ in sequence avoiding duplication. It keeps the descriptive texts and allows searching in the titles. File cabinets are used to store paper documents.

Electronic documents are managed in a version management system, which stores differences between versions of a document and allows the reconstruction of previous versions (we use subversion, abbreviated to SVN).

The data are, for security, stored on a mirrored disk accessible over the web, using an OBBC access to the database. Regular (yearly) backups on a storage medium kept in a different location guarantees long term availability; transfer of the database and the SVN archives from the current software to future solutions will be necessary. For documents storage as PDF files is strongly encouraged to allow at least reproduction in the future.

The use of the three different components, i.e., GZ database doucument version management SVN and file cabinets, but especially the coordination between the GZ databases and the SVN archive of electronic documents required care and experienced showed that discrepancies crept in.

Hamster is an approach to combine the two components under a joint control. The goal was a simplification for the user, primarily in the management of electronic documents in the team integrated with the lookup of GZ for search

in the file cabinets. The development served for me as a non-trivial example of interactive software coded in Haskell with a graphical user interface.

2.3 Basic Concepts

The unit is the GZ, which has a unique number (the GZ proper)

- and a descriptive text, which should contain any term by which one later may expect to find the GZ (avoid umlaut and other special characters!),
- a number of paper documents collected in a folder,
- an electronic folder (with subfolders) with electronic datasets (files),
- the unique GZ number is used to connect the pieces. Therefore, all folder names must start with the GZnumber. The remainder of the folder names can be decided individuelly and is not relevant, (e.g., 4879 Hamster).

3 User

Each member of the team can create a GZ and enter documents. Please describe the GZ with all terms that could possibly be used to search for the GZ; if later other terms become relevant, add them to the descriptive title. This manual focuses on the treatment of electronic files and assumes the mechanism of standard filing and ordering, which are well known.

A user local copy of a folder is initially inserted. later folders are checked out by users and revised versions are written back (committed); concurrent work is detected by SVN but conflicts must be resolved manually. The user's folder that are treated by GZ and SVN should all be kept in one directory, called the user's work directory (in principle a user can have several work directories and could work with several SVN archives, but this is not recommended and requires advanced understanding of SVN details). Examples of recommended practice are given later.

4 Installation

- A working installation of an SVN server ; required information is the URL of the SVN server (svn://gi28.geoinfo.tuwien.ac.at/gz), a valid user name and password.
- A database with the database schema necessary (appendix 1); on the user's computer an ODBC connector must be set up, which needs the username and password.

This gives the data sources, where GZ and SVN will connect to. It is customary to keep a shortcut to the executable of GZ and SVN in a convenient location. GZ and SVN stores a preference file in a hidden file ".svngzPrefs.txt"; if this

file is not found, a new file is initialized to access test versions of DB and SVN. To start the SVN and DB, the server must be available (to start the initial preference file, the test server). The preference file is a text file and can be edited; the syntax is regular Haskell syntax.

Error message: If the server indicated in the preferences file is not available the program stops responding, waiting for ever ("hangs").

5 Usage

The screen is divided by a column of command buttons on the right side of the screen and information fields on the left. The command buttons activate the corresponding commands (See 5.2.1 to 5.2.13). The information fields are divided in fields where user input is expected (keywords, GZ, title) and the other fields, where only information from the system is communicated to the user. The size of the fields is automatically adjusted to the window size and the amount of information to display. A new user can perform the following actions:

4.1 FIND a GZ by entering a keyword the GZ title might contain.

4.2 GET the description to a given GZ (and all other information about the corresponding files in the local workspace and in the SVN repository).

4.3 CREATE a New GZ with a descriptive text.

4.4 CHANGE the descriptive text for a GZ.

4.5 Given a GZ CHECKOUT the most recent versions (earlier versions require direct user of an SVN client). A hint for the descriptive part of the filename can be entered.

4.5 ADD all new file to a GZ (confirm with commit).

4.6 Make NEW svn folder in the repository. Change the descriptive text for a GZ.

4.7 MERGE files from a local folder to the SVN folder.

4.8 INSERT all files in a folder in the workspace into the SVN archives for a new GZ.

4.9 REVERT (undo) the additions.

4.10 Enter and change the user data, workspace location, etc.

4.11 CLEAR the screen <check correspondences with buttons>.

4.12 EDIT the user data, workspace location, etc.

5.1 Edit Preferences

The command EDIT opens a second small screen to enter prefernces. The system starts the first time with a standard (test) setting for the preferences. These can be edited with the "Edit Prefs" button. Enter the correct values and click "OK" (or "cancel" to not change anything).

Changes of the database have no immediate effect.

5.2 Commands

5.2.1 Find GZ

Enter keywords in the keywords field and hit either the "FIND" button or simply return (while the censor is in the keywords field). You can add keywords if your first search gets to many GZ or delete some if not found what you searched for. Keywords are combined by AND (i.e., they must all be present in the title of the GZ). The result is shown in the information field that adapts its size.

Nothing found gives empty information field and an information in the status box.

If only a single GZ matches your keywords then it is shown in the GZ field: <not yet done: missing code for "one found">.

If several GZs match, select with the cursor from the list and click to enter the GZ number in the GZ field and a "GET" action performed (see 4.3).

5.2.2 GET information about a GZ

Enter a GZ (or have one entered previous from a FIND operation) and GET will retrieve all information available about this GZ:

- title from database,
- the directory name in the SVN,
- the directory name in the local work directory,
- the files in the SVN directory if it exists (and their correspondence with the local files) or just the local files not yet in the SVN (where the GZ does not exist).

Error:

- more than one folder in the work directory has this GZ. Must be corrected by user.
- more than one folder in the SVN has their GZ. Report!
- the superfolder in the SVN does not exist, if this was not caused by an old (predating our use of SVN) or an invalid GZ entered, then report.

5.2.3 INSERT a folder

Given a folder in the workspaces, insert all its content in the SVN repository for the GZ. The current folder is renamed to "inserted ..." and can be deleted. To continue working, the stored folder must be CHECKED OUT. It is checked that for this GZ no folder has been previously created.

5.2.4 CHANGE title of GZ

If you use a GZ and observe that the title does not include a keyword you would use to retrieve this GZ, add it in the field and press CHANGE. The title is updated and it will be possible to find the GZ with this new additional keyword.

5.2.5 CHECKOUT

the folder in SVN presently shown is copied to your workdir and you can use the files.

5.2.6 NEWSVN

If you created a GZ for which electronic documents should be stored, make a new folder in SVN. Next, you may want to check out this empty folder to put files into it. NEW SVN requires a short descriptive filename.

5.2.7 INSERT an existing folder

If you have already a folder for this GZ (with a name starting with the GZ) in your work directory and no folder exists in SVN; then use this command to enter all files. Please clean up the folder and delete unnecessary files (.turp, .bak,). Your original folder is renamed with a prefix "inserted". If you want to continue on this GZ, you must CHECKOUT the GZ.

5.2.8 MERGE your files into an SVN folder

If you have files in a folder for which an SVN folder exists already, use this command to merge the files. Files for which you have a different context as in repository are put in a folder "previous content" and need your inspection. If they are updates to the ones currently in SVN copy, the one is the GZ folder. Additional (new) files you must ADD to SVN. At end commit all changes.

5.2.9 UPDATE your workdir

If somebody else has worked and commited a new version of a document to the repository, use UPDATE to bring all your files in the GZ folder in your working directory to the newest state.

5.2.10 ADD all files to SVN

New files you enter in a GZ folder are not automatically inserted into the repository, because we want to avoid collecting unnecessary files (e.g., -.bak, -turp, or ~files). Write this you add all new files, marked with a "?" in the SVN field.

Easier and more selective is the shortcut: which on a file marked with "?" as new and it is added (make changes to "A"). Additions become definitive only after COMMIT.

5.2.11 REVERT

This command changes all files marked for addition ("A") back to new ("?").

Easy shortcut: click on a file, marked for addition ("A") and only this file is reverted back to new ("?").

5.2.12 COMMIT

This command brings your changes from your local copy to the repository. It adds all files marked for addition ("A") and ahnges the files you have modified (mark "M") with each COMMIT you should enter a comment what your workstep was doing; COMMIT often – it adds only the difference between the versions to the repository, but you can then always go back to the last commited version (with SVN operations, not in Hamster).

5.2.13 CLEAR ALL

This serves only to clear all entries on screen to avoid confusion. A clear slate to start a new task!

5.2.14 EDIT preferences

Opens the secondary screen. Closes this screen to continue working!

5.2.15 QUIT

Stops Hamster; your changes to preferences are written to disk and avoidable for the next !

5.2.16 CREATE GZ

Inserting a title in the title field and click NEWGZ. Please enter all . A new GZ is entered into the DB with this titles and a new GZ returned, but not yet into the SVN repository (use NEW Folder).

5.2.17 New SVN FOLDER

For a new GZ, you will collect electronic documents to create an SVN folder. Put a hint for the filename (some short description text) and click NEWSVN-FOLDER: a folder with the name GZ followed by your hint is created:

- If you need it in your local directory, you must CHECK it OUT.
- Error: a folder with this GZ exists already. You can check it out.
- ADD: If you have created new files in a folder, which is checked out, you can add all of them with the command ADD. That you need to add is visible by "ß" at the beginning of lines listing the files in your work folder.
- Adding single files: to add a single file, click on the line in the of SVN status in the repository, which shows a question mark "?".
- Commit: to communicate the changes in files and files you added to the repository, enter a desthen over criptive comment for the character of your completed work step and click COMMIT.

5.2.18 CHANGE title of GZ

If you use a GZ and observe that the

5.3 Error possible:

The folder in your work directory was not checked out from this repository (or the filename in the repository changed by manual intervention). Do

1. rename work folder by preceding the GZ with some character (e.g., "My 4435 XXX").
2. checkout the current folder from the repository. This gives a new folder "4435 xxx" in your work directory.
3. copy all files from "My 4435 xxx" to the new "4435 xxx" (overwriting!).
4. GET should show your changes.
5. Add or add single files.
6. COMMIT the folder "My 4435 xxx" can be deleted.

6 Best Practice

Here I describe a few typical tasks and how they are best solved:

6.1 Start a New Task

1. Search with FIND to make sure no GZ exists.
2. Enter a descriptive title and CREATE a new GZ.
3. make a NEW SVN folder (enter a short title)
4. CHECKOUT this folder .... work and store your files in the GZ folder ...
5. GET Hamster updated
6. ADD the files (best individuallly by clicking on them)
7. COMMIT changes with comment on your work step

If the task is finished you can move the GZ folder from your workdir to the directory where you keep work done. It can later be brought back (or and save, checked out).

6.2 Work on a GZ Others Have Already Worked on

1. FIND GZ by keyword
2. CHECKOUT the folder into your workdir ... work ... FIND or
3. GET to see what is changed
4. ADD new files (individually)
5. COMMIT with comment on what you have done

6.3 You Have a Folder to Insert in the Reprository

FIND GZ or CREATE one. If an SVN folder exists already, use MERGE, if not, you can simply INSERT (and CHECKOUT if you want to continue working).

7 Dialog and Errors

The operations check that all required inputs are given. If something is missing, the status box says invalid inputs. In general, the statusbox shows messages when a command reached an unusual result (e.g., nothing found) but the user can notice by, e.g., a field that is blank. If an ALERT box is shown and gives a hint for corrective actions. A message box is shown for errors that should not occur. To improve Hamster, please take a screenshot of the task window an the message box and send it to me by email; I will try to fix the problem.