Guildford Protocol

Current Maintainer: Thomas Krichel

1 Introduction

This document is the Guildford protocol. It is named after the town where it has been written. The protocol provides a set of rules for the publication and exchange of documents on the internet. It could be implemented in any group that wishes to distribute documents on the internet.
The idea behind the protocol goes back to a statement by William L. Goffe. On 15 July 1995, he wrote on the (now defunct) NetEc-admin list:

> What I would suggest is this: a distributed system with any number of sites, each mirroring each other. It would have extensive bibliographic functions (cross-referencing, etc.), and my favorite, digital timestamps for when the papers were put up. For archives outside it, papers could be listed, but no cross-referencing. But, such archives could "join" the system (say it was written in perl so could run on NT as well as Unix). Then you'd have the best of both worlds: distributed, anybody could join, extensive cross-referencing, the whole works. Such a system could easily grow with the profession's use of the net. Such a system would GREATLY benefit the profession.

The way to achieve this "global and local" archive is through a comprehensive distribution process that is based on a set of archives. An archive is a machine that makes data available. It is a place where original data enters the system. The data are then distributed to any number of sites. A site is a collection of archives on the same computer system. It usually consists of a local archive augmented by frequently updated copies of remote archives. The local archive is maintained on the local computer, whereas the remote archives are maintained on other computers. We call a frequently updated copy of one archive on a remote site a "mirror". There is no need for every site will need to mirror every archive in the system. Some may only mirror bibliographic information rather than the papers to conserve disk space. Others may mirror all the files of an archive. Others will mirror only parts of a few archives.
All archives hold papers and metadata about papers, as well as software that is useful to maintain archives. Everything contained in an archive may be mirrored. For example, if the full text of a paper is in the archive, it may be mirrored. If the archive does not wish the full text to be mirrored, it can store the papers outside the archive, for example on a directory that does not belong to the archive.
The Guildford protocol aims to find a set of minimal restrictions on archives such that a global and local system will work. A second key aim of this document is to provide a set of rules such that if they are followed locally, require almost no central effort. However a small amount of work has to be provided by a central archive. This archive is called the core archive. It contains ReDIF templates that describe all known archives in the system.
A limitation of this document is that it will not deal with charging money for metadata i.e. that the description of documents is free. Limiting the access to the documents themselves is possible but remains outside of the scope of the document.
A second limitation is that the protocol does not deal with archiving and preservation issues. A key feature of the protocol is that each document has exactly one home site. If the home site withdraws the document it is withdrawn (after a short delay) on all site that maintain copies of the document.
A third limitation of this document is that it applies only to archives that contain series of documents. It is not intended to apply to homepage style publications.
The last limitation is that the protocol is not concerned with providing end user services. For example the protocol does not provide any ideas on how to present documents on a web server, how documents should be indexed etc. However one of the key features of the protocol is that software used to perform these task can be written by a community of contributors and distributed among sites for the benefits of everybody. There may be a variety of software tools written to render the data in all sorts of ways, to prepare all sorts of indexes, fire up a number of search tools etc. At the moment many archives are

trying hard to tackle these issues on their own for their local needs. The usage of the protocol will go a long way towards eliminating this multiplication of effort.

2 Definitions and conventions

The "authority" is a group of people that have come together to implement the Guildford protocol on a set of documents and metadata. "RePEc" is an authority that supports the creation and deployment of ReDIF data in economics according to the Guildford protocol. For the rest of the document, we will use the name "RePEc" to refer to the authority.
"ReDIF (Research Document Information Format)" is a set of rules to encode information about papers, series, and copying rights. It is discipline independent and independent of an organisational structure that supports its creation and deployment.
A "series" is a collection of documents that are kept together.
An "archive" is a directory on a computer that is open to access by ftp or http. It holds a collection of series of papers or a collection of data about papers held elsewhere.
A "RePEc archive" carries data formated in ReDIF that pertain to Economics and use RePEc as administrator. It may also carry the full text of the documents.
A "site" is a collection of (normally one) local archive plus any number of mirrored archives. For the purpose of identification, sites and archives are treated identical. Normally each site runs on archive and mirrors several others.
The "core archive" is a single machine on which a limited number of important files are kept. These include the documentation of the protocol, including the regulations of the decision making process, brief descriptions of all the software contributed to the organisation and the core templates (see the ReDIF documentation) of all participating archives. The contents of the core site is made available on all other sites.
The "administrator" is the person who keeps the core files on the core site.
A "site" is a local archive complemented by mirrored copies of remote archives. It offers end-user facilities.
"mirroring" is a process by which copies of series of documents are made from one site to another, such that the contents of the archive is the same on all sites except for a short delay that is inevitable.

3 Introduction to implementation

Two pieces of software that are crucial for the implementation of the project are the ftp or/and http protocol. The data is stored on sites that are accessible via either those protocols, but we recommend to use ftp. To manage the sites, we have developed perl scripts. Perl is available freely for a variety of platforms but it most popular on unix machines. Initially the software will be designed with unix machines in mind. We will consider other platforms when they become available.
To enable the software to work, we will need to impose a minimum of structure on archive. That is detailed in Section 3.1. For the provision of software for all servers, we need a fix some rules that are detailed in Section 3.2.
The central site ALL operates at ftp://netec.mcc.ac.uk/pub/NetEc/RePEc/all. Its archive code is ALL. It provides the mirror and the webcopy perl scripts that are used to mirror, the former for ftp archives, the latter for the http archives. For ftp archives ftp://netec.mcc.ac.uk/pub/NetEc/RePEc/all/conf/allarch.mir provides a file that mirrors all the ftp archives. ftp://netec.mcc.ac.uk/pub/NetEc/RePEc/all/docu contains all the documentation. Finally ftp://netec.mcc.ac.uk/pub/NetEc/RePEc/all/ mirrors the core templates of all archives.

3.1 The site

A site contains a set of files. All names of files are case-insensitive. The convention is that all archive identifiers have three letters, all series identifiers 6 letters, the reserved words have four letters. Archive identifiers are awarded by RePEc, the series identifiers are fixed by the archives in consultation with RePEc, and the reserved words are those mentioned in the protocol. All files ending with the extension `.rdf`, pronounced "ReDIF" are files that contain ReDIF templates.
If a site runs on a multi-user machine, it is recommended to create a special account "adrepec" for the account. The following files can then live in the home directory of the user "adrepec", or better, in a subdirectory `RePEc` of this account that points to a space in the files system that is accessible via anonymous ftp. In the following we assume that we are in this subdirectory and we are looking at the files and directories that it contains.
`./`*archive_identifier*`/`*archive_identifier*`arch.rdf` a file describing the archive using a single ReDIF archive template (mandatory)
`./`*archive_identifier*`/`*archive_identifier*`seri.rdf` a file describing the series in the archive using a sequence of ReDIF series templates. All series in the archive are described in this file, one template for each series. (mandatory)

`./archive_identifier/archive_identifierserv.rdf` a file describing the services that the site is offering
`./archive_identifier/archive_identifiermirr.rdf` a file describing the mirroring arrangement of the site, using the ReDIF-mirror template
`./archive_identifier/series_identifier/` a directory wfor papers and metadata for the series that is identified by *series_identifier* are stored. All files that that pertain to the series *series_identifier* must be stored in that directory. There must be at least one series directory on each server. Files that contain ReDIF information are called ReDIF files. Their names must end in `.rdf`, but otherwise the structure of the directory is free. You may put all templates for all papers in the series in one file or you may put each paper template in a different file, just do as you please. It is good pratice to start each ReDIF file in the directory with the *series_identifier*.
./*archive_identifier*`/inst/` a place to store ReDIF files that contain institution templates. See the ReDIF draft for further informaiton about that template.
./*archive_identifier*`/soft/` software that is written locally. For example, an archive may wish to write a specific procedure by which its ReDIF-Paper data is translated into html.
./*archive_identifier*`/conf/` location of configuration files for software. It does not matter whether the software is supplied by the local archive or a remote archive.
`./remo/`*archive_identifier*`/` This is a strong suggestion (in the sense that some software may not work if you to not follow it) where to put information from a remote archive *archive_identifier*. A site does not need to mirror all the files from a remote archive. It must mirror the archive templates of the remote archive. It may then select series to mirror, or only mirror the `rdf` files.
For any archive, the collection of `./`*archive_identifier*`/`*archive_identifier*`????.rdf` where *????* is either `arch`, `seri`, `serv` or `mirr` are called its core templates. They are mirrored on the core site.
Each site may mirror a number of series from any number of archives. If any site mirrors any series for an archive, it may mirror the complete subdirectories of the series or all ReDIF files (ending with `.rdf`) of the series. If an archive does not wish the papers to be mirrored, then it will store them out of this hierarchy.

3.2 Providing software

Archives may provide software for other archives to use. Services are encouraged to provide software that allows the construction of identical services on many sites.
For the sake of simplicity of exposition, let us assume that there is an archive "giv" that gives (provides) a software package called "bambi", and an archive "tak" that takes (uses) the software locally. A software packages is a set of software files that together perform a particular function. All software is written in perl. All scripts assume that perl lives in `/usr/bin/perl` to allow direct use. Thus all scripts starts with `#!/usr/bin/perl`.
Software is provided in the `./giv/soft` directory. Software written at the local site will be found in the `./giv/soft/RePEc` directory, all other software is placed in other subdirectories of `./giv/soft`. For example, the ALL archive provides mirror and webcopy, two sets of scripts to mirror ftp and http based archives respectively. These scripts are used in scripts provided by ALL.
The file `./giv/soft/RePEc/bambi` is the main executable. Names of other files start with 'bambi' (e.g. `bambi.README`) or be placed in directory `./giv/soft/RePEc/bambi/` in order to distinguish which file belongs to which packages.
Scripts used by various programmes are called `./giv/soft/RePEc/soft/giv.*` or are placed `./giv/soft/RePEc/soft/giv/`
All scripts at start perform some general procedures to ensure their portability (between different sites and environments) and general rules of usage.
They first check for command-line options `-rdir` *directory_name* and `-conf` *filename*) to allow user to set her or his RePEc-archive directory (which ends with archive identifier) and chose a programm configuration file name (instead of default) respectively. The latter is not necessary if programme does not use a configuration file.
If `-rdir` option is not given, then programme checks environment variable `REPECDIR` for RePEc-archive home directory. If it is not set, then it assumes, that current directory is RePEc directory of the local archive. Keep in mind that some Perl functions in your script may require absolute path of RePEc directory, thus a simple `./` may be too simple.
If RePEc-archive home directory was received through a command-line option or as environment variable, it should be checked whether such a directory exists at all, and whether it ends with a three letter sequence, which is supposed to be the local archive identifier. Under Unix scripts should be aware of the optional trailing backslash.
In ftp://netec.mcc.ac.uk/pub/NetEc/RePEc/all/soft/RePEc/guilp_use_perl.eg we show you just an example of Perl code to implement this algorithm, which was written by Ivan Kurmanov. You are welcome to adapt it in your programmes. Ivan welcomes comments and suggestions.

Some scripts need to be installed before use by the end user. In that case the installation script is `./giv/RePEc/bambi.in`, again with a configuration example `./tak/bambi.in.conf`. The installation script could be run again each time the software has been mirrored or each time an service provider wishes to update the version of the script in use at the archive. The latter should be a more secure way to update scripts.