

Directory Services and WWW integration using SOLO

Jean-Christophe TOUVET

Paul-Andre PAYS

INRIA
Domaine de Voluceau, Rocquencourt
78150 Le Chesnay Cedex
France

Abstract

The starting point of the work presented here, with our background in directory services, was when we started to really use the Web the strong feeling that WWW was really lacking a name resolution protocol in addition to all already existing information search and retrieval functionalities. On the other hand the directory services and White Pages Services we were involved with were lacking a good integration with the available information services.

The new SOLO protocol, that we have contributed to define, aims at providing the directory and WP services with several highly desirable features such as simplicity, technology independence, user-friendly naming and a centroid-like indexing scheme. One of our goals with SOLO was to be able to use SOLO to meet the previous paragraph requirements.

SOLO protocol is so simple that it took us only a few weeks to implement gateways between SOLO and X.500 or WWW. The next step was the integration of this protocol directly into WWW clients by defining a new URL type `solo://host/query` and adding a very simple module in the common library (*libwww*). Hence WWW would be supplemented with a built-in access to most directory and WP Services.

WWW has proven so successful that there are already billions of documents available over the net. Faced with these enormous amount of information the weak part is the lack of nicely integrated structured and indexed catalog services. We are convinced that the integration of Directory oriented protocols will be very helpful at contributing to solutions.

Furthermore, as SOLO is relatively well suited for name server type operations, it could be a good candidate to provide URN/URL resolution, or at least, as the URI topic is hot and difficult, at enabling experimentation with proposed approaches and solutions.

A first version of this development already exists at writing time and we plan to make it widely available before the conference. In the mean time, you can test our prototype SOLO/WWW gateway.

Keywords: World Wide Web, WWW, SOLO, directory services, URL, URN

Table of Contents

- 1. Directory services and the Web
 - 1.1. White Pages
 - 1.2. Distributed document catalogs
 - 1.3. URN → URL resolution
- 2. The SOLO approach
 - 2.1. Protocol description
 - 2.2. Hypertext fitness
 - 2.3. X.500 access
- 3. Implementation
 - 3.1. Client side
 - 3.1.1. Gateway approach
 - 3.1.2. SOLO integration
 - 3.2. Server side
 - 3.2.1. Which servers ?
 - Index servers
 - SOLO data servers
 - X.500 servers
 - 3.2.2. URLs and directory services
 - URL as directory attributes
 - URL referring to directory entries or attributes
- 4. Examples and Demonstration
 - 4.1. SOLO/WWW gateway
 - 4.2. Integrated access
- 5. Conclusion
- Acknowledgments
- References
- Acronyms

1. Directory services and the Web

As of today, there is no distributed Directory Service directly accessible from the Web. All we have is emerging X.500 gateways and local databases searching through gateways or WAIS. What Directory Services could contribute to the Web?

1.1. White pages

There is clearly a need for a global WPS in the Internet. As WWW becomes the most popular Internet service, we think that its browsers *will* be amongst the privileged human interface to this WPS, thus solving one of the identified major problems for WPS: the lack of well accepted and largely deployed User interfaces.

Conversely, we know that people need to give access to WWW objects from their Directory entry (Home Page, Photo, Project description...). Simply including a WPS client into a WWW browser provides an easy solution.

As SOLO follows the X.500 data model, all the ongoing X.500 efforts and developments are available, without excluding any other directory technology nor putting any barrier to creativity in the fast evolving directory domain. On the other hand our solution enables solves many of the X.500 identified problems or weaknesses.

1.2. Distributed document catalogs

Today, there are very interesting projects to manage lists of Web servers in the world, but as their number increases amazingly, we think that centralized databases will soon become inadequate to store informations about all documents contained into these servers.

All this requires one or a few widely accepted naming schemes, a name to locators resolution mechanism, hence directory services. Of course this does not exclude all the indexing which is partly covered today by the indexing part of WAIS, but comes as another mandatory building block.

What is needed to face this growth is a distributed database mechanism, delegating document indexing at organizational level. A solution would be to widely deploy directory services including of course X.500 as it is today the only well-defined and **working** solution. Here again SOLO is a good solution, as it can transparently be used as a native service or as a simple front-end to X.500.

1.3. URN → URL resolution

Disclaimer: We certainly don't pretend here to solve the URI, URN, URC, URL issue, thus please understand URN in this context only as a name for resources or documents available on the net.

Documents, files and more generally resources need to have a name with amongst many other the following characteristics: stability over time and independence of the locations of available instances. As soon as we have a name a resolution mechanism is required.

There is no doubt about the need for a directory service to perform this task. Today, most proposals are focused on DNS usage; in this schema, the URN has two parts: – *a domain name*, – *an identifier*. Thus, the DNS could be used to get the adequate resolution server address for this domain name, then the resolution server, from the document identifier, could return the URL for the document. We can see here two weak points:

- *protocol between domain server and Web client has to be defined,*
- *absolutely no link between DNS and a distributed searching mechanism as described above (since DNS has no searching capabilities).*

Our proposal, while not pretending at all to be the only and ultimate solution, will show a more integrated approach, using the same protocol to map URNs into URLs **and** to search these documents.

2. The SOLO approach

SOLO design came from several years working in the directory services area and tries to solve some of the shortcomings of the existing solutions, especially but not exclusively with the aim of deploying an Internet wide White Pages Service. SOLO naming and data model is largely derived from X.500 but aims at being implemented with many existing or still to be defined data-base technologies.

SOLO can be, at the same time, considered as

- a complete directory service
 - as it includes an access protocol and a distribution and navigation mechanism and as servers can be build on top of any existing or proprietary local data-base mechanism.
- a complement to existing distributed directory services
 - to which it brings its simple access protocol and a powerful indexing mechanism.
- a mere access protocol
 - to an already existing directory service

The description below is based on examples, for more precision of the protocol please use the draft RFC.

2.1. Protocol description

To describe SOLO, we must separate two different issues:

- Client/Server protocol
- Servers internetworking

SOLO access protocol

The Client/Server dialog is really simple, thus a few examples will suffice (*nota: there only one request: SOLO, in the examples below, all lines not commencing with SOLO are answers from the server*)

```
% telnet champagne.inria.fr solo
Trying 128.93.2.15 ...
Connected to champagne.inria.fr.
Escape character is '^]'.

SOLO <huitema,inria,fr> ? email;
500 Matches: <CN=Christian HUITEMA,O=inria,C=fr>
Email: "Christian.HUITEMA@inria.fr"
.

SOLO <silvestre,recherche,france> ? CN, Address, Email, Loc;
202-No such name: <silvestre,recherche,france>
301-Partial Match: <C=FR>
400-Suggestion: <CN=SILVESTRE,O=cnrs,C=fr>
400-Suggestion: <CN=SILVESTRE,O=RENATER,C=fr>
400 Suggestion: <CN=SILVESTRE,O=inria,C=fr>

SOLO <CN=SILVESTRE,O=inria,C=fr> ? CN, Address, Email, Loc;
202-No such name: <CN=SILVESTRE,O=inria,C=fr>
301-Partial Match: <O=inria,C=fr>
400 Suggestion: <CN=Peter SYLVESTER,O=inria,C=fr>

SOLO <CN=Peter SYLVESTER,O=inria,C=fr> ? CN, Address, Email, Loc;
500 Matches: <CN=Peter SYLVESTER,O=inria,C=fr>
CN: "Peter SYLVESTER"
Email: "Peter.SYLVESTER@inria.fr"
L: Rocquencourt
.

SOLO <Class=Document+SOLO,inria,fr> ? Author,CN,Desc,SeeAlso;
202-No such name: <SOLO,inria,fr>
301-Partial Match: <O=inria,C=fr>
400-Suggestion: <CN=SOLO-WWW paper,O=inria,C=fr>
400 Suggestion: <CN=SOLO Internet Draft,O=inria,C=fr>
```

```

SOLO <CN=SOLO-WWW paper,O=inria,C=fr> ? Author,CN,Desc,SeeAlso;
500 Matches: <CN=SOLO-WWW paper,O=inria,C=fr>
Author: <CN=Jean-Christophe Touvet,O=inria,C=fr>,
        <CN=Paul-Andre Pays,O=inria,C=fr>
CN: "SOLO-WWW paper"
Desc: "Directory Services using SOLO integration into WWW",
      "WWW 94 conference submitted paper"
SeeAlso: <CN=SOLO Internet Draft,O=inria,C=fr>
.

QUIT
602 Closing connection.

```

We can note some goodies from these examples:

- Very simple TCP connection and client/server dialog
- User Friendly Naming
- Easy-to-parse answers

SOLO indexing and distribution mechanism

Servers internetworking mechanism uses the WHOIS++ centroid concept: indexing servers will hold gigantic tables in order to manage navigation. Once a SOLO server is set up, its manager sends a SGNL query to one (or more) indexing server, then the latter connects to the new server to POLL its indexed attributes. Each time the content of its base is changed a SOLO data server needs to SGNL the index server(s) so that they can POLL it again.

```

% telnet pax.inria.fr solo
Trying 138.96.24.78
Connected to pax.inria.fr.
Escape character is '^]'.

SOLO <solo,inria,fr>? Email;
302-Suggested-server: <O=inria, C=fr> champagne.inria.fr
302-Suggested-server: <O=inria, C=fr> solo.enst.fr:8990

QUIT
602 Closing the connection.
.

```

The SOLO server pax.inria.fr contains plenty of indexes, obtained by a POLL mechanism, which enables it to return to the client queries one or a set of suggested servers which are likely to be able to answer the query. It is up to the client to decide to make use of this information and to query the suggested server(s).

2.2. Hypertext fitness

We've seen in previous examples that when an entry can't be matched exactly, the server returns *suggestions*. Our experience showed that this mechanism fits perfectly into an hypertext browser: "just click on the suggestion".

Entry results, with "Attribute: comma-separated-list-of-values" are easy to parse, and it is also very simple to create attributes containing URLs, which are used to create hypertext links.

2.3. X.500 access

SOLO was designed to be compatible with X.500, and it's in fact quite easy to build a SOLO front-end to X.500.

In addition to providing X.500 with an Ultra Lightweight Access Protocol, SOLO firstly offers a solution to possibly huge attributes by enabling these attributes to be stored and/or returned as URLs instead of full values and equally brings a centroid based indexing alternative to X.500 heavyweight distribution and navigation proprietary scheme.

3. Implementation

3.1. Client side

On the client side, we considered and implemented two solutions:

- Standard Web clients using HTTP to dialog with a SOLO/HTTP gateway
- Full integration of the SOLO protocol in the clients

3.1.1. Gateway approach

As said before, this solution is cheap and easy and has been developed in a matter of weeks, in fact it was in parallel with the SOLO specification and it contributed to help in the SOLO design. On the other hand, gateway mechanism isn't as efficient as direct integration of the protocol. In fact, if you want to contact SOLO server X, you have to set up a connection with gateway Y which connects itself to server X. Another weakness lies in the URLs syntax, since you have to provide both gateway and SOLO server in it, as:

```
http://champagne.inria.fr:8889/mitsou.inria.fr:8888/...
```

Anyhow a good point with gateways is that they provide the service to clients which don't have the capability by themselves. It is still the case with some *aging* Mac browser with WAIS. It enabled us to have many people experimenting with the prototype with their usual browser, while we were not willing to freeze and distribute the new *libwww* module.

3.1.2. SOLO integration

Integration of SOLO into the WWW common library *libwww* was not really difficult. In fact, such a module is smaller than the Gopher one.

Since SOLO can easily be built as a front-end to X.500, it has also been possible to use this module to resolve x500 : URLs. We provide this service using an environment variable or a file which contains the address of a SOLO front-end to X.500.

This also leaves the door open for URN -> URL resolution, resolving urn : scheme. In practice, for our current experiment we use the same SOLO access server than for x500 : URLs, which means that we use X.500 to store URN and related informations (URC and URL).

3.2. Server side

3.2.1. Which servers ?

Having a gateway or a nice client module for SOLO, what kind of servers are we able to connect to ? The answer is easy: any SOLO capable server. In the current experiment we already have

Index servers

These are centroids returning suggestions (SOLO speaking: suggested servers) about servers likely to be able to answer the query. The suggestion is based on the index base that this server has collected by polling many other SOLO servers.

SOLO data servers

These are the servers really holding directory entries. The actual data-base technology is completely transparent. In practice such a server can be as simple as a the public domain one from Christian Huitema which interfaces a NIS or Unix Password like flat file, or as powerful and complex a front-end to a traditional data-base such as Oracle

X.500 servers

In such a case the solo server acts as a front-end to a X.500 DSA, which indirectly provides access to the whole distributed directory. The implementation can vary from a simple SOLO server including an X.500

DUA (our prototype) to a much more integrated SOLO capable X.500 servers (under development by at least one X.500 supplier: TS-E3X for its UCOM.X product line, because they are also developing SOLO capable Email user agents).

3.2.2. URLs and Directory Services

Another issue for which we are proposing a solution is the use of URLs in conjunction with a directory.

- URL as directory attributes

White Pages entries

Any people directory entry, by means of URLs can now refer any resource as an attribute, which is by far a much better and scalable solution than having these resources directly held by the directory itself. It is better in term of performance, in term of consistency and more generally in terms of integration with the complete information system.

Document entries

Without pretending to solve the URI URN issue, it is extremely easy to allocate a name to any network reachable document (ie. any piece of retrievable information on the net), to register a directory entry which contains plenty of related information as attributes (such as authors, title, version, references, date) and also provides one or more locators (URL) of the known retrievable instances.

- URL referring to directory entries or attributes

Conversely, with this approach, it is easy and useful to designate any directory held information as a URL. By having a SOLO capable Web document authors are now in a position to have hyperlinks pointing to resources stored in the directory. It can be either a complete directory entries or even one or more attributes of a given directory entry. Examples of one entry and one attribute URL:

```
solo://champagne.inria.fr/%3Cpaul-andre pays,inria,fr%3E!
solo://champagne.inria.fr/%3Cjean-christophe touvet,fr%3E!Phone
```

If now we have a URL which refers to a directory entry which happens to be the entry for a document, we have a perfect integration of URN and URLs

```
solo://champagne.inria.fr/%3Class=Document+CN=Solo RFC,inria,fr%3E!
```

is a **SOLO URL**, but is indeed **also** a (form of) **URN** It can be **now** used as an hyperlink in any Web document and which does **not** require any additional concept nor software nor service to be perfectly operational.

From this, it was trivial to define (no joke) the following type of URL:

```
urn:%3CCN=Solo RFC, O=inria, C=FR%3E
```

which is not different from the previous one, except that the choice of the server is left to a local environment variable, and that *Document Class* is implicit. In our prototype it is interpreted by using the very same solo server.

URN exist and are usable now!. We are very conscious that this radical approach which somehow blurs a maybe artificial distinction between URN and URL might not be well received by the WWW or IIA community because it may add confusion to the already confuse enough URN issue. We do not pretend to meet all the requirements that have been expressed in the IETF URI group. We however consider it is of importance to really explore this track and to experiment with it, because it is available today and solves many problems.

4. Examples and demonstration

4.1. SOLO/WWW gateway

In order to illustrate all the above and our current developments let's use the Web and its possibilities through the **examples** included below.

Considering that as of May '94 we have already entered the Web era, we apologize that non Web readers will miss most of the interest of the next few pages and we invite them to use as soon as possible a form capable WWW browser.

To give an idea of the implementation itself, it suffices to say that the current gateway is written in Perl and is small.

Basic access to the gateway: champagne.inria.fr:8889

SOLO examples

- Searches

One exact match

[<huitema,inria,fr>?*](#)

Approximate matches and suggestions

[<christian huitema,inria,fr>?*](#)

[<peis,inria,fr>?*](#)

[<tou*,inria,fr>?*](#)

- Other possibilities

User-Friendly Country Naming

[<huitema,inria,france>?*](#)

Very approximate !

[<robb*,*x*,england>?*](#) (hint: try Nexor LTD)

Organization and country entries

[<O=inria,fr>?*](#)

[<C=gb>?*](#)

And what about European Pilots ?

[<OU=password,value*,europe>?*](#)

Aliases and multiple RDNs

[<O=nexor,us>?*](#)

Local context

[<huitema>?*](#) same as [<huitema,O=inria,C=fr>](#) for this server

- Absolute matches (name server type queries)

Person entry

[<CN=christian huitema,O=inria,C=fr>!*](#)

Organisation entry

[<O=inria,C=fr>!*](#)

Error

[<CN=nobody,O=inria,C=fr>!*](#)

- Document searches

Document entry

[<Class=Document+CN=Solo-WWW paper,O=inria,C=fr>?*](#)

With default UFN attributes scheme

[<Class=Document+Solo Internet Draft,inria,fr>?*](#)

Searching

[<Class=Document+Solo,inria,fr>?*](#)

X.500 Searches

With a gateway, you have to specify an X.500 access server within the "http" URL.

Multiple matches

[champagne.inria.fr:2222/<peis,inria,fr>?*](http://champagne.inria.fr:2222/<peis,inria,fr>?)

Organisation entry with absolute match

[champagne.inria.fr:2222/<O=inria,C=fr>!*](http://champagne.inria.fr:2222/<O=inria,C=fr>!)

Document URNs

You must also provide in the "http" URL the mapping server.

URN resolution

champagne.inria.fr:2222/<CN=Solo-WWW paper,O=inria,C=fr>!URL

With default UFN attributes scheme

champagne.inria.fr:2222/<Solo Internet Draft,inria,fr>!URL

NOTE: When attribute types are not specified the default following UFN scheme is used:

CN, (OU,)* O, C

4.2. Integrated access

And now a very similar set of examples but using a SOLO capable browser. In order to be able to follow the hyperlinks included in this page it is required to use a WWW browser which has been linked with the new *libwww* module that we have developed for Solo. Solo integration enables the processing of the following URL new families:

```
solo://host/query
x500:query
urn:query
```

In the last 2 cases there is no specified server (nor host). This reflects the **globality** of these URLs which are in fact URNs. Of course an environment variable must indicate which SOLO server to use, which is the second case **must be** a X.500 front-end.

The results are the same as with the gateway except for better performance and mostly the ability to use much simpler and cleaner URLs.

5. Conclusion

The conclusion will be very short.

We hope that this paper and mostly the included demonstrations were convincing enough of what a relatively small investment was able to bring to the Web and to WPS and to directory services in general.

The key word here is certainly that:

- directory services are a requirement for all informations services.
- the ability to handle URL is a requirement for directory services
- a nice integration of these, as provided by SOLO, is a requirement of the **users** and their applications.

We consider this as a starting point as many issues has not been addressed at all, especially in the URN domain for which this proposal aims only to be a building block. Much remains to be done.

And, last but not least: we had fun!

Acknowledgments

Many thanks to many people, Tim Berners Lee who originated the Web, the ones who contributed to improve and deploy it, Christian Huitema who initiated the SOLO design and a few years before X.500 technology, the TS-E3X people who provided us with the X.500 products and also contributed to the SOLO specifications, all our INRIA CSR/Comiso team coworkers who had to work harder on the daily tasks while we were having fun with this design and implementation.

Jean-Christophe.Touvet@inria.fr, Paul-Andre.Pays@inria.fr

References

Simple Object LOk-up protocol (SOLO)
 Internet Draft, draft-huitema-solo-00.txt, INRIA TS-E3X
 C. Huitema, P-A. Pays, A. Zahm, A. Woermann, December 1993

Hypertext Markup Language (HTML)
 Internet Draft, draft-ietf-iiir-html-01.txt
<http://info.cern.ch/hypertext/WWW/MarkUp/MarkUp.html>
 Tim Berners-Lee, July 1993

HTML+ (Hypertext markup format)
 Internet Draft, draft-raggett-www-html-00.txt
 Dave Raggett, October 1993

Protocol for the Retrieval and Manipulation of Textual and
 Hypermedia Information (HTTP)
 Internet Draft, draft-ietf-iiir-http-00.txt
<http://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTTP2.html>
 Tim Berners-Lee, July 1993

Uniform Resource Locators (URL)
 Internet Draft, draft-ietf-uri-url-01.txt
<http://info.cern.ch/hypertext/WWW/Addressing/Addressing.html>
 Tim Berners-Lee, October 1993

Universal Resource Identifiers in WWW
 Internet Draft, draft-www-uri-00.txt
http://info.cern.ch/hypertext/WWW/Addressing/URL/URI_Overview.html
 Tim Berners-Lee, March 1994

What WWW needs from WAIS and X.500
 OSI-DS Draft, <ftp://info.cern.ch/pub/www/doc/wais-x500-w3.ps>
 Tim Berners-Lee, March 1992

Using the OSI Directory to achieve User Friendly Naming
 Internet RFC 1484 / OSI-DS 24
 Steve Hardcastle-Kille, July 1993

The Directory --- overview of concepts, models and services
 CCITT X.500 Series Recommendations, December 1988

Future of X.500 report: RFC-1588.
 John Postel et al, January 1994

Acronyms

DAP: Directory Access Protocol
DIT: Directory Information Tree
DNS: Domain Name System
DSA: Directory System Agent
DSP: Directory System Protocol
DUA: Directory User Agent
HTML: HyperText Markup Language
HTTP: HyperText Transfert Protocol
SOLO: Simple Object LOk-up protocol
UFN: User Friendly Name
URL: Uniform Resource Locator
URN: Uniform Resource Name
WPS: White Pages Service
WWW: (also known as the **Web**) World Wide Web