A Common LISP Hypermedia Server

John C. Mallery
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA USA

Keywords: HTML+, Interactivity, Technology, Servers.

Hypertext: http://www.ai.mit.edu/projects/iiip/doc/cl-http/server.html

Menu

Motivations

1. **Dynamic HTML+ generation.**
2. **Generalizing email form-processing.**
3. **Seamless operation over multiple protocols.**
4. **High-productivity programming in LISP.**
5. **Interfacing AI research systems.**
6. **Exploring interactive hypermedia.**
7. **Linking AI and WWW communities.**

Themes

- **Application Level:**
  1. **Artificial Intelligence systems.**
  2. **Public access to government.**
- **World-Wide Interface:**
  1. **Interactivity through form-processing.**
  2. **Flexible interaction through dynamic generation of HTML+.**
  3. **Presentation types enable direct manipulation of data.**
- **Standards Implications:**
  1. **Rich & Extensible presentation system for HTML+.**
  2. **Transfer of presentation types, queries, and forms.**
- **Programming Technology**
  1. **HTML+ authoring functions.**
  2. **Object-oriented implementation.**
  3. **Handling exceptional conditions.**
  4. **Rapid prototyping.**

Applications

- **Document Search:** Retrieve government documents via a taxonomy of categories used to distribute them over email.
- **Public Access Email:** Subscribe to electronic publications through a forms to an email server.
- **Electronic Publications:** Subscribe to electronic publications through a forms interface to an email server.
- **Natural-Language Question Answering:** Ask English questions about the M.I.T. Artificial Intelligence Laboratory.
- **Rule Induction:** Learn *if-then* rules about international conflict management since 1945.

Server Features

- **HTTP Methods:** GET, HEAD, POST (form processing).
- **Network Security:** Subnet security is based on client IP address.
- **Export Types:**
  1. **Text Types**: Text-File, HTML-File, LISP-File, Text-Directory, LISP-Directory, HTML-Form, HTML-Computed.
  2. **Other Types**: X-Bitmap-Image, GIF-Image, JPEG-Image, Basic-Audio, MPEG-Video, Postscript-File.
  3. **Redirect**

Server Design

- **Object-Oriented:** Key data structures are implemented as CLOS objects (server instances, URLs, errors) and major procedures as generic multimethods.
- **Typed Condition Handling:** Common LISP Condition classes are defined for all HTTP response codes and are arranged hierarchically.
- **HTML+ Authoring Functions:** A set of operators allows programs to generate HTML+ dynamically.
- **HTML+ Input Types:** Tags for HTML+ input fields are consolidated in an abstraction analogous to CLIM presentation types.

Classes of HTTP URLs

- **Path URLs** map to a directory hierarchy, or if desired, a physical file.
- **Object URLs** denote specific documents with a name, and usually an extension.
- **Search URLs** specify searches on databases and may have an associated HTML document to describe the search.
- **Form URLs** have a *response function* to handle POST operations, but handle GET just like object URLs.
- **Computed URLs** fully compute their response to a GET operation.

Common LISP Interface Manager

- **Presentation Types:**
  1. **Present Method** displays an object to the user.
  2. **Accept Method** parses and validates user input.
- **Inheritance based on type hierarchy.**
- **Presentation translators** map between LISP objects based on their associated presentation types.
- **User input accepted more intelligently** because more equivalences can be resolved by means of type subsumption and presentation translation.
- **Simpler yet more powerful interface model** because bidirectional translation allows users to *directly* manipulate program data while perceiving only external, user-friendly representation.

HTML+ Authoring Functions

- **Functional Interface for generating HTML+.**
- **More Compact Vocabulary:** *e.g.*, enumeration and text-style macros.
- **Input Types:**
  1. **Implemented as CLOS classes (like CLIM).**
  2. **Accept Method:** Writes appropriate HTML+ tags and parses returns during form submission.
  3. **Standardized Parameters across most types.**
  4. **Input types:** Audio, date, float, image, integer-range, multi-line-text, multiple-choice, radio-button, reset-button, select-choices, string, submit-button, url.
- **Query Abstraction:** Enables atomic questions.
- **Smooth Code Evolution:** thanks to an abstraction layer.

Computed URL Example

```
(export-url "http://clinton.ai.mit.edu/radio-addresses.html"
            :html-computed
            #'compute-clinton-audio-page)

(defmethod compute-clinton-audio-page ((url url:http-computed-url) stream)
   (flet ((write-pathname (pathname stream)
           (let ((url (url:pathname-ftp-url-string pathname))
                 (friendly-date (uu-radio-address-anchor-date pathname)))
             (when friendly-date
               (html+:enumerating-item (stream)
                 (html+:with-rendition (:bold :stream stream)
                   (html+:note-anchor
                     friendly-date :reference url :stream stream)))))))
    (let ((directory-list (www-utils:ftp-directory-info
                           *saturday-radio-addresses-audio-directory*
                           "anonymous"
                           (www-utils:user-mail-address))))
      (if directory-list
          (with-successful-response (stream :html)
            (html+:with-document-preamble (:stream stream)
              (html+:declare-base-reference url :stream stream)
              (html+:declare-title "Saturday Radio Addresses" :stream stream))
            (html+:with-document-body (:stream stream)
              (html+:with-section-heading
                ((html+:image
                   "http://clinton.ai.mit.edu/icons/sound-icon.xbm"
                   "The President's Saturday Radio Addresses to the Nation"
                   :stream nil))
                (html+:new-paragraph :stream stream)
                (write-string "By clicking on a date, you can listen to
                              the audio for a Saturday radio address
                              by President Clinton." stream)
                (html+:horizontal-line :stream stream)
                (html+:new-paragraph :stream stream)
                (html+:with-enumeration (stream :itemize)
                  (loop for (pathname) in directory-list
                        do (write-pathname pathname stream))))))
          (error "Unable to connect to UU.NET to get file list.")))))
```

Communications Linker System

- **Email Servers:** Message routing, automatic surveys, interface to arbitrary computations.
- **Document Routing and Retrieval:** *Document universes* use taxonomies of categories to index documents for retrieval and routing via boolean combinations of categories.
- **Automatic Form Processing:** User interaction via textual forms exchanged in email.
- **Graphical Form Authoring Tool:** The interface allows users to define automatic surveys and forms (Renaud, 1994).
- **Meta-level Form Processing:** Automatic retries and form/query exchange become possible.

Automatic Form Processing

- **Forms:** Composed of a series of *queries*.
- **Query:** Has an associated CLIM presentation type to present its value and parse any user input.
- **Parsing Forms:** Convert textual input into an internal representation.
  1. **Scan**: Locate the query between delimiters.
  2. **Intern**: Convert the query name into the internal query object.
  3. **Accept**: Parse text following the delimited query using the accept method for the query's presentation type.
  4. **Handle Errors**: If the query value is invalid, signal an invalid input condition.
- **Successful Parse:** Execute the form's response function using the parsed query values.
- **Parse Failure:** Retry the form by returning to the user a form with correct values defaulted and an explanation about how to fix invalid input.

Conclusions

- Complex LISP systems from AI can be directly interfaced to the World-Wide Web.
- Interactive hypermedia applications and protocol extension can be rapidly prototyped in LISP.
- COMLINK will soon process forms seamlessly over email and HTTP.
- A basic set of presentation types can be tested on the server side.
- Presentation types for client support can be proposed.
- Transfer of definitions to clients needs to be explored.