Human-Computer Interaction

Sergio Luján Mora
sergio.lujan@ua.es
@sergiolujanmora
CARD SORTING
INFORMATION ARCHITECTURE AND INTERACTION DIAGRAM
• A visual vocabulary for describing information architecture and interaction design, by Jesse James Garrett
THE ELEMENTS OF USER EXPERIENCE
USER-CENTERED DESIGN FOR THE WEB

Jesse James Garrett
AJAX
The Elements of User Experience

A basic duality: The Web was originally conceived as a hypertextual information space; but the development of increasingly sophisticated front- and back-end technologies has fostered its use as a remote software interface. This dual nature has led to much confusion, as user experience practitioners have attempted to adapt their terminology to cases beyond the scope of its original application. The goal of this document is to define some of these terms within their appropriate contexts, and to clarify the underlying relationships among these various elements.

Web as software interface

Visual Design: graphic treatment of interface elements (the "look" in "look-and-feel")

Interface Design: as in traditional HCI: design of interface elements to facilitate user interaction with functionality

Information Design: in the Tuftean sense: designing the presentation of information to facilitate understanding

Interaction Design: development of application flows to facilitate user tasks, defining how the user interacts with site functionality

Functional Specifications: "feature set": detailed descriptions of functionality the site must include in order to meet user needs

User Needs: externally derived goals for the site; identified through user research, ethno/techno/psychographics, etc.

Site Objectives: business, creative, or other internally derived goals for the site

task-oriented

Web as hypertext system

Visual Design: visual treatment of text, graphic page elements and navigational components

Navigation Design: design of interface elements to facilitate the user's movement through the information architecture

Information Design: in the Tuftean sense: designing the presentation of information to facilitate understanding

Information Architecture: structural design of the information space to facilitate intuitive access to content

Content Requirements: definition of content elements required in the site in order to meet user needs

User Needs: externally derived goals for the site; identified through user research, ethno/techno/psychographics, etc.

Site Objectives: business, creative, or other internally derived goals for the site

information-oriented

This picture is incomplete: The model outlined here does not account for secondary considerations (such as those arising during technical or content development) that may influence decisions during user experience development. Also, this model does not describe a development process, nor does it define roles within a user experience development team. Rather, it seeks to define the key considerations that go into the development of user experience on the Web today.
Software Interface: querying database
Hypertext system
Lublin

From Wikipedia, the free encyclopedia

For other uses, see Lublin (disambiguation).

Lublin [ˈlublijn] (listen) (Ukrainian: Луцьк; Yiddish: בָּלוֹבָּן; Polish: Lublin) is the ninth largest city in Poland and the second largest city of Lesser Poland. It is the capital of Lublin Voivodeship (province) with a population of 349,103 (March 2011). Lublin is the largest Polish city east of the Vistula River. Lublin is approximately 170 kilometres (106 miles) southeast of the capital, Warsaw.

Contents [hide]
1 History
   1.1 Jagiellonian Poland
   1.2 World War II
   1.3 Solidarity years
2 Climate
3 Population
4 Economy
   4.1 Media
5 Transport
   5.1 Airport
   5.2 Roads
6 Culture and tourism
   6.1 Museum
   6.2 Cinema
   6.3 Theatres
   6.4 Galleries
   6.5 Old Town
   6.6 Pubs and restaurants
   6.7 City of festivals
7 European Capital of Culture
8 Education
9 Sports
10 Notable residents
11 Politics
   11.1 Lublin constituency
12 International relations
12.1 Twin towns — sister cities
13 See also
14 References
15 External links

Flag
Cost of arms


Software Interface
Hypertext system

Lublin

From Wikipedia, the free encyclopedia

For other uses, see Lublin (disambiguation).

Lublin ([ˈlublin] (listen) (Ukrainian: Лу́блин, Liublin; Yiddish: וויבּילין; Hebrew: לובּלין) is the ninth largest city in Poland and the second largest city of Lesser Poland. It is the capital of Lublin Voivodeship (province) with a population of 349,103 (March 2011). Lublin is the largest Polish city east of the Vistula River. Lublin is approximately 170 kilometres (106 miles) southeast of the capital, Warsaw.
A visual vocabulary
for describing information architecture and interaction design

version 1.1b (6 March 2002)

Jesse James Garrett
(contact)

Translations of this document are available:

- Chinese (thanks to Arky Tan)
- Japanese (thanks to Atsushi Hasegawa)
- Spanish (thanks to Javier Velasco)
- Italian (thanks to Laura Caprio and Beatrice Ghiglione)
- French (thanks to Francois Lamotte)
- German (thanks to Marcus Brinckhoff)
- Portuguese (thanks to Livia Labate and Laura Cretton Lessa)

Table of Contents

1. Summary
2. Version history
3. Initial considerations
4. Conceptual underpinnings
1. **Project sponsors and managers** use them to obtain a general sense of the scope and form of the project.

2. **Content producers** use them to derive content requirements.

3. **Visual and interface designers** use them to derive a count of how many unique page designs must be produced, and to obtain an initial sense of the navigational and interface requirements for these designs.

4. **Technologists** use them to derive functional requirements.

5. **Information architects and interaction designers** use them to develop detailed navigational and interface requirements for each page.
Goal

• To guide a (team of) designer or (team of) programmer who would produce the website
Requirements

• **Whiteboard-compatible:** The vocabulary should be simple enough that diagrams can be sketched quickly by hand. The elements of the vocabulary should be distinct enough from each other that moderately sloppy drawing cannot compromise the clarity of the diagram.

• **Tool-independent:** The vocabulary should be designed so that specialized software tools are not required in order to construct diagrams. The vocabulary should not favor the use of any particular software tool, but should instead enable architects to work with the tools they are most comfortable using.
Requirements

• **Small and self-contained:** Because the diagrams are used by a diverse range of audiences with differing levels of knowledge of (or even interest in) diagramming systems used in other areas of technical development, the vocabulary should not require such knowledge or interest. The total set of elements should be kept as small as possible, maintaining a strict one-to-one correlation between concepts and symbols, so that the vocabulary can be learned and applied quickly. The concepts expressed by a diagram may be arbitrarily complex; the means of their expression should not be.
Level of detail

• The diagram focuses on the **macrostructure**, providing just enough detail to enable team members to get the "big picture"

• The task of the architect is to determine the appropriate level of detail to meet this objective

• The specific page-level detail, or **microstructure**, is detailed in other documents that the architect may not be primarily responsible for developing
ELEMENTS
Page

- The page is a unit of *presentation*, not (necessarily) a unit of implementation -- one page in your diagram may correspond to multiple HTML files (as in a frameset interface) or multiple units of code (as in a server-side include or database-driven implementation)
File

• A resource without navigational properties
• Files are delivered to the user for use outside a Web browser environment (such as audio or video files, stand-alone documents like PDFs, or executables)
Stack

• Pagestack: a group of functionally identical pages whose navigational properties are immaterial to the macrostructure of the site

• Filestack: a group of files that receive identical navigational treatment and can be classified as a single entity (such as a collection of downloadable games or a library of PDF instruction manuals)
**Figure 1a:** [left] The page and the file

**Figure 1b:** [right] The pagestack and the filestack
Labels: These don't need to correlate to actual designations such as HTML `<TITLE>` elements or filenames, but they do need to be unique for each page or file in the diagram.

Figure 1a: [left] The page and the file
Figure 1b: [right] The pagestack and the filestack
RELATIONSHIPS
• Relationships between elements are depicted with simple lines or **connectors**. These conceptual relationships will inevitably translate into navigational relationships -- but not all navigational relationships will appear in the diagram.
Figure 2a: *[left]* A simple tree structure

Figure 2b: *[right]* The same structure as in 2a, diagrammed differently
• **Arrows**: convey *directionality* to indicate how the user will move through the system toward completion of a particular task
• Arrows are not like the arrows indicating a one-way street, but rather like the arrows indicating the way to the food court in the mall.

• The user is not prohibited from moving in the opposite direction; the arrow merely indicates the direction in which the user is likely to want to go.
Figure 3a: [top left] Arrow indicates downstream movement toward task completion

Figure 3b: [bottom left] Crossbar indicates upstream movement is not permitted

Figure 3c: [right] Multiple arrowheads clarify directionality
**Labels:** the use of these should be limited to cases in which the action taken by the user needs to be clarified. If the labels become long and unwieldy and start to clutter the diagram, point the reader toward a footnote or appendix entry.

**Figure 4a:** *left* A superfluous label
**Figure 4b:** *middle* A useful label
**Figure 4c:** *right* A footnote or appendix reference
OTHER ELEMENTS
**Concurrent set**: a user action generates multiple simultaneous results (such as spawning a pop-up window at the same time a page is loaded in the main window, or displaying a page while a file is being downloaded)

*Figure 5: A concurrent set*
Continuation point: list one or more sources or destinations as needed.

**Figure 6a: [left]** A "continue to" point refers the reader to another diagram

**Figure 6b: [right]** A "continue from" point, picking up where 6a left off
**Area:** used to identify a group of pages that share one or more common attributes (such as appearing in a pop-up window, or having some unique design treatment). Use labels to identify these attributes or (as with connectors), refer to notes elsewhere in the document if you have a lot to say.

---

**Figure 7:** An example use of an area to represent a pop-up window
**Iterative area**: repeating the same basic structure as it is applied to a number of functionally identical information elements.

*Figure 8*: An example use of an iterative area to represent a repeated structure in a product catalog.
Reusable components: flow areas and references

Figure 9a: [left] A flow reference serves as both a "continue to" point and a "continue from" point

Figure 9b: [right] The flow area referenced in 9a
CONDITIONAL ELEMENTS
Figure 10: An example use of a decision point in a login sequence

Decision point
**Conditional connector:** is used when a path may or may not be presented to the user depending upon whether one or more conditions are met.

**Figure 11a:** [left] A conditional connector

**Figure 11b:** [right] A conditional arrow
**Conditional branch**: the system must select one path among a number of mutually exclusive options to be presented to the user

![Diagram of a conditional branch](image_url)
**Conditional selector** element: functions much like the conditional branch, with one important difference: with the selector, the various downstream paths are not mutually exclusive. Any number of the paths that fulfill the condition(s) may be presented to the user.

![Diagram](image.png)

*Figure 13: A conditional selector*
**Cluster**: the cluster can appear downstream from either a conditional branch or a conditional selector.

*Figure 14: A cluster downstream from a branch*
**Conditional area**: when one or more conditions applies to a group of pages

*Figure 15: An example use of a conditional area where a secure connection is required*
<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
</tr>
<tr>
<td>Page Stack</td>
</tr>
<tr>
<td>File Stack</td>
</tr>
<tr>
<td>Continuation Points</td>
</tr>
<tr>
<td>Connectors</td>
</tr>
<tr>
<td>Conditional Connectors</td>
</tr>
<tr>
<td>Area</td>
</tr>
<tr>
<td>Iterative Area</td>
</tr>
</tbody>
</table>

**Visual Vocabulary Quick Reference**

- **Page**
  - Pages are the fundamental unit of presentation on the Web, not necessarily a unit of implementation. One page in your diagram may correspond to multiple HTML files (as in a frame set interface) or multiple units of code (as in a server-side or database-driven implementation).

- **File**
  - Files are parcels of data without navigational properties. They are delivered to the user for use outside a Web browser environment (such as audio or video files, stand-alone documents like PDFs, or executables).

- **Page Stack**
  - A page stack indicates a group of functionally identical pages whose navigational properties are immaterial to the macrostructure of the site.

- **File Stack**
  - A file stack represents a group of files that receive identical navigational treatment and can be classified as a single entity (such as a collection of downloadable games or a library of PDF instruction manuals).

- **Continuation Points**
  - Continuation points allow us to separate our diagrams into easily digestible sections. We use continuation points to bridge the gaps between sections. A single continuation point may list one or more sources or destinations as needed. The choice of orientation is a matter of the architect's aesthetic judgment.

- **Connectors**
  - Relationships between elements are depicted with simple lines. Connectors use arrows to convey directionality, indicating how the user will move through the system. We use a crossbar on the opposite end of the arrow to prohibit upstream movement.

- **Conditional Connectors**
  - A conditional connector is used when a path may or may not be presented to the user depending upon whether one or more conditions are met. Conditional connectors use arrows to convey directionality, indicating how the user will move through the system. We use a crossbar on the opposite end of the arrow to prohibit upstream movement.

- **Area**
  - An area is used to identify a group of pages that share one or more common attributes (such as appearing in a pop-up window, or having some unique design treatment). Use labels to identify these attributes or (as with connectors), refer to notes elsewhere in the document if you have a lot to say.

- **Iterative Area**
  - Iterative areas are used to represent architectures that involve repeating the same basic structure as it is applied to a number of functionally identical information elements. For example, you may have a product catalog in which each product has a number of pages associated with it.

- **Conditional Area**
  - A conditional area is used when one or more conditions apply to a group of pages. Conditional areas are applied most commonly in situations involving access permissions, such as when a valid login or encrypted (SSL) connection is required. Conditional areas are associated with a result generated in the event that the condition(s) are not fulfilled.

- **Flow Area**
  - A flow area encloses a sequence of steps (like a login procedure, for instance) that will appear repeated in different contexts throughout the design, analogous to a programming procedure. Flow areas require the use of two special types of continuation points: entry points and exit points.

- **Flow Reference**
  - A flow reference serves as a sort of "placeholder" for a flow in every context in which it is repeated. Both the flow area and flow reference elements have the same basic shape, a rectangle with the corners clipped off.

- **Concurrent Set**
  - A concurrent set is used in cases where a user action generates multiple, simultaneous results (such as spawning a pop-up window at the same time a page is loaded in the main window, or displaying a page while a file is being downloaded). Upstream elements connect to the curved side; downstream elements connect to the flat side.

- **Decision Point**
  - A decision point is used to model when one user action may generate one of a number of results, and the system must make a decision about which result is to be presented. Note that arrows must be used in conjunction with decision points to clarify whether associated elements are upstream or downstream from the decision point.

- **Conditional Branch**
  - A conditional branch is when the system (not based on user action) must select one path among a number of mutually exclusive options to be presented to the user. Upstream elements connect to one point of the triangle; downstream elements connect to the opposite side.

- **Conditional Selector**
  - Conditional selectors function much like the conditional branch, with one important difference: with the selector, the various downstream paths are not mutually exclusive. Any number of the paths that fulfill the condition(s) may be presented to the user (e.g. search results.)

- **Cluster**
  - A cluster is used when a system can present more than one path based upon certain conditions. The cluster can appear downstream from either a conditional branch or a conditional selector. For example if the attribute being evaluated has value x, the user sees a path to page C; but if the attribute has value y, the user sees paths to both page C and page D.
EXAMPLE
NOTES
(1a) If user is logged in, return edit user prefs. If user is not logged in, return login.
(1b) If user is logged in, return post new topic. If user is not logged in, return login.
(1c) Display links to topics posted in the last n days, where n is defined in user prefs. For users not logged in, n=7.
(1d) Display links to topics matching search criteria.
(1e) Display links to topics posted in selected month.
(1f) If user is logged in, logout function is available.

Jesse James Garrett
http://www.jig.net/ia/visvocab/
entry point: home

entry points:
home
decision (1a)
decision (1b)

login/register

login

register

registration confirmed

login confirmed

exit point:
edit user prefs

exit point:
home

NOTES
(2a) If login info is valid, return login confirmed. If login info is invalid, return login.
(3a) Functionality for the MetaTalk area is not documented in this diagram.
TOOLS
Other shape libraries available:

- Stencil file for Visio 2000
- Stencil file for Visio 5
- Stencil file for Visio 4
- PowerPoint file (thanks Fredrik Johansson Oviedo for some good suggestions)
- Library file for Adobe InDesign 2.x (thanks Andrew Robinson)
- Library file for Adobe InDesign 1.x
- Library file for FreeHand 10 (thanks Andrew Crow)
- Library file for FreeHand 9 (thanks Andrew Crow)
- Illustrator EPS file
- Library file for iGrafx Flowcharter 2000 (thanks Andrew Robinson)
- Library file for OpenOffice (thanks Nelson Rodriguez-Peña)
- Collection of EPS files containing one element per file, suitable for import into other applications (1.1 MB)
- Two different SVG libraries: [1] (1.9 MB) [2] (52 KB; thanks to Marcus Brinkhoff)

© 2000-2002 Jesse James Garrett
This document: http://www.jjg.net/ia/visvocab/
More IA resources: http://www.jjg.net/ia/