

Aclaración de ideas erróneas sobre AI

El arquitecto de información es un miembro vital de los equipos de desarrollo Web, y juega un papel esencial en la organización del contenido de los sitios. El objetivo de éste artículo es aclarar algunas ideas erróneas sobre la arquitectura de información y ayudar a definir el papel que juega el arquitecto de información en el desarrollo de los sitios Web.

Por Thomas Myer. tom@myerman.com

Traducido por Jorge Arango. <http://www.jarango.com/es/>

¿Qué es exactamente la arquitectura de información? ¿Es lo mismo que diseño? Algunas descripciones laborales parecen sugerir que la profesión incluye desde administradores de bases de datos, hasta diseñadores visuales y escritores técnicos.

En la práctica, el arquitecto de información prepara una gran parte de los cimientos para la organización del contenido de un sitio, sin importar dónde resida el contenido -- ya sea en archivos planos, multimedia, o campos en una base de datos. Y en sitios complejos y portales, el arquitecto de información ayuda a evitar que la falta de organización convierta la experiencia del usuario en una pesadilla.

En éste artículo voy a contestar algunas de las preguntas planteadas en la introducción -- aclarando algo de la confusión existente -- y voy a proveer algunos recursos para lectura adicional.

Piensa arquitecto, no diseñador

La mejor manera de comprender la diferencia entre un arquitecto de información y un diseñador es comparar la diferencia entre un arquitecto de edificios y un diseñador de interiores.

Al arquitecto le interesa principalmente la estructura, flujo, y otros aspectos fundamentales del edificio, como lo son la localización de los sistemas de plomería y electricidad. Si el arquitecto no hace su trabajo, el edificio puede caerse, o -- en un caso menos extremo -- no satisfacer las necesidades de las personas que lo habitan. Por ejemplo, puede resultar que el edificio no tenga suficientes recámaras.

Por su parte, los diseñadores de interiores están interesados en el color, la localización, y el estilo de los muebles, las texturas, y las superficies -- en otras palabras, el atractivo sensorial del edificio. Pueden estar tratando de impartir un aspecto específico a las recámaras (por ejemplo, Mediterráneo o Español), o asegurarse que ciertos colores y estilos estén presentes consistentemente a través de toda la estructura.

Ésto no quiere decir que uno u otro de estos trabajos sea más fácil o más difícil. Simplemente son diferentes. Obviamente, siempre van a solaparse un poco (por ejemplo, el arquitecto esta interesado en el atractivo visual, y el diseñador esta interesado en el flujo y acceso del edificio), pero por lo general las dos disciplinas se complementan. Usted no le pediría a un diseñador de interiores que le diseñara la estructura de una casa, y probablemente no le pediría a un arquitecto que opinara sobre el esquema de colores de las paredes de su residencia.

Existen paralelos en el campo de desarrollo Web. El arquitecto de información por lo general no es experto en diseño de identidades, colores, composición, y ciertas formas de comunicación visual. Éstos campos son la especialidad del diseñador. Sin embargo, el arquitecto de información conoce de categorización, XML, creación y organización de contenido, diseño de interactividad, y diseño de navegación. Su experiencia radica en las estructuras informáticas. El resto de éste artículo presenta cuándo, dónde, y cómo se puede aplicar ésta experiencia.

Cuándo y dónde trabajan los arquitectos de información

El mejor momento para involucrar a un arquitecto de información es al inicio del

desarrollo del proyecto. Al igual que uno no llamaría a un arquitecto tradicional a una construcción en el momento de techar conjunto de paredes inestables, es preferible llamar a un experto antes de afrontar problemas con un sitio Web.

Sin tener en cuenta en qué parte del proceso se involucre a un arquitecto de información, él va a querer comprender los siguientes parámetros del proyecto:

- Metas y necesidades de los usuarios
- Metas del negocio u organización
- Limitaciones tecnológicas
- Limitaciones del contenido
- Limitaciones del proyecto

El concepto de entender los objetivos y necesidades de los usuarios no debe sorprender a los profesionales de usabilidad que lean éste artículo. Sin embargo, algunas de las otras ideas pueden ser nuevas para el grupo que sigue a Jakob Nielsen. El principal objetivo de un arquitecto de información es construir estructuras informáticas que sean fáciles de entender y utilizar. Para poder lograrlo, debe alinear las metas del negocio con los objetivos de los usuarios, mientras trabaja dentro de las limitaciones del proyecto (fechas de entrega, recursos, presupuesto, etc.), asunto técnicos (lenguaje de desarrollo, base de datos, etc.), y producción del contenido (personal disponible, conocimientos del talento interno o subcontratado, etc.).

Por ejemplo, la selección de un lenguaje de programación por encima de otro puede afectar la manera en que se maneja el estado de sesión en un portal de contenido, lo cual puede afectar la interacción básica entre los usuarios y el contenido. Específicamente, puede causar que el contenido sea desplegado en una página larga única en vez de dividirse en varias páginas más pequeñas. Otro buen ejemplo de limitaciones del contenido puede ser el caso de un pequeño grupo interno de escritores con poca (o ninguna) experiencia sobre el tema abarcado por el sitio. Ésta limitación puede contradecir la principal meta del negocio: crear un sitio con profundidad editorial sobre ése tema particular. En éste caso, el negocio puede considerar contratar más escritores o incrementar el presupuesto para escritores subcontratados.

Aunque ambos asuntos parecieran no tener mucho que ver con la organización y despliegue de información en el sitio, la interacción básica del usuario con el contenido puede verse profundamente afectada.

El contenido de un sitio Web no es un objeto estático: puede cambiar no solo de estructura y medio de despliegue, sino también de un día a otro. Para lograr una arquitectura de información efectiva, es esencial recordar la estrategia de contenido y mantenerlo alineado con los objetivos del negocio.

Cómo trabajan los arquitectos de información

El proceso de la arquitectura de información ocurre usualmente en dos pasos: primero, usando una metodología de arriba a abajo, seguido por una de abajo a arriba.

La investigación de arriba a abajo usualmente ocurre primero. Como sugiere su nombre, éste método ayuda al arquitecto de información a comprender todo lo que abarca el proyecto, y le permite elaborar los detalles del despliegue, flujo, y estructuración del contenido.

Es posible comprender las metas de los usuarios y la organización, pero es necesario también traducir éstas metas en una imagen general de la estructura de contenido del sitio. Por ejemplo, si el sitio va a ser un portal para la venta de equipo de nieve e información al respecto, el arquitecto de información debe comprender cómo piensan los compradores y usuarios de éstos equipos acerca de éste contenido. En pocas palabras, debe comprender el "espacio conceptual" del usuario. Por ejemplo, los snowboards y esquís pueden constar de diferentes categorías de productos e información. Y el

esquí puede contar con diversas sub-categorías, como el esquí montañés y competitivo. La forma que toma la estructura del sitio de arriba a abajo ayuda a hacer más navegable el contenido.

A continuación presentamos el método de abajo a arriba. Éste método ayuda al arquitecto de información a comprender los pedazos pequeños que hacen que todo encaje, como la metadata y todos los asuntos relacionados a la metadata: qué debe hacer, dónde debe estar almacenada, cómo utilizarla, y cómo interactúan los diferentes tipos de metadata.

La metadata es esencial para gran parte de la actividad en el sitio, desde la búsqueda, hasta la personalización y organización del contenido. ¿Qué es la metadata? Como sugiere su nombre, es literalmente “data acerca de la data.” Por ejemplo, cuando hablamos, las palabras que salen de nuestra boca pueden considerarse como data. El tono de nuestra voz, nuestro lenguaje corporal, e inclusive la personalidad y actitud de nuestro interlocutor pueden considerarse metadata. La metadata enriquece nuestra comprensión de la data.

Para seguir con el ejemplo anterior del sitio de venta de equipo de nieve, digamos que los dueños del sitio quieren que cualquiera que lea un artículo sobre esquí competitivo puedan ver en un recuadro productos relacionados en la tienda virtual del sitio. Una manera de crear éste efecto es crear una taxonomía de metadata -- un tipo de jerarquía de términos y conceptos que pueden ser asignados tanto al contenido como a los productos. Por ejemplo, una taxonomía parcial para el sitio puede verse así:

Snowboards

Esquíes

Esquí competitivo

Esquí montañés

Esquí slalom

Botas de nieve

Etc.

Si se crea y usa una taxonomía centralizada, no es necesario adivinar cuál es la relación entre los productos y el contenido al momento de hacer vínculos entre los mismos. Si todos los productos y el contenido están localizados en el lugar correcto en la taxonomía, los usuarios que ven un artículo marcado como esquí competitivo van a ver productos de esquí competitivo en el recuadro.

Además, si un año después los dueños del sitio deciden implementar un sistema de personalización, la misma taxonomía puede ser utilizada. Los usuarios pueden crear cuentas y escoger qué tipo de contenido y productos quieren ver o enfatizar dentro de la lista ofrecida por la taxonomía. Los problemas se reducen debido a que la misma taxonomía se está utilizando para la personalización, los productos, y el contenido. Los usuarios que muestran afinidad por el contenido y productos relacionados al snowboarding pueden recibir todos los artículos marcados como tal.

Cómo reciben su información

Los arquitectos de información deben recopilar mucha información dentro de períodos de tiempo cortos. Para lograrlo, utilizan métodos como éstos:

1. Solicitando listas prioritarias de funcionalidad deseada, ya sea por medio de email, en persona (“uno-a-uno”), o en sesiones de grupo. Éstas listas son efectivas porque permiten a los usuarios tener sesiones creativas sin obstáculos o limitaciones. Dar prioridad a las funciones deseadas ayuda a administrar el filtrado de los requisitos y deseos. Una lista sintetizada, visible a todos los participantes, ayuda a dar una idea general de la escala del proyecto y permite asignar responsabilidad del proceso al grupo entero.

2. Conduciendo sesiones en el tablero con grupos pequeños que permiten a los participantes explorar diferentes áreas conceptuales. Estas sesiones pueden utilizarse para hacer creatividad pura, dar seguimiento al ejercicio anterior (las listas), o componer los componentes y el flujo de las páginas. Si es cierto que una imagen vale por mil palabras, ver el proceso de creación de la imagen (en sentido literal y figurativo) vale varios miles de palabras.

3. Elaborando un análisis competitivo de 3-5 sitios similares al del cliente, y presentando los resultados de dicho análisis. Este ejercicio no solo puede ayudar a desenterrar requisitos y deseos adicionales, sino que también puede ayudar a identificar la funcionalidad base para la organización. Por ejemplo, si el sitio de un competidor cuenta con 500 artículos, la administración de su compañía puede determinar exceder esa cantidad al final del trimestre siguiente. Desde el punto de vista del usuario, el incluir funcionalidad popular en el sitio -- aquellas cosas que han llegado a esperar como funcionalidad base de los otros sitios -- podría ser útil. Y por supuesto, evitar el mal diseño y decisiones arquitectónicas hechas en otros sitios puede mejorar el sitio que su equipo está construyendo.

4. Los análisis de tareas y audiencias son métodos clásicos para averiguar quién es la audiencia, y qué esperan lograr los usuarios del sitio. El análisis de audiencia puede comprender desde la investigación demográfica y psicográfica, hasta estudios etnográficos de usuarios reales. El resultado final de este trabajo por lo general consiste de casos que definen usuarios específicos (también conocidos como actores) y sus tareas. Por ejemplo:

Pedro administra un equipo de desarrolladores de software en una corporación global. Tiene 30 empleados en su departamento, y necesita las noticias más recientes sobre lenguajes de programación. No está interesado en los aspectos técnicos de los mismos, sino en el impacto que tienen las decisiones tecnológicas sobre las prácticas del negocio, particularmente en la disminución del tiempo que toma llevar el producto al mercado.

5. Los grupos focales de usuarios son una buena manera de obtener comentarios y reacciones acerca de los prototipos o diseños del sitio. Si quiere usar esta técnica, asegúrese de usar un moderador con experiencia, alguien que pueda obtener buena información sin influenciar a los usuarios. También asegúrese de estar al tanto (formal o informalmente) de las pruebas que se hacen al grupo focal. Las observaciones y grabaciones usualmente se hacen fuera de vista del grupo, con buena razón. Cualquiera que sabe que está siendo observado naturalmente cambia sus reacciones y comportamiento.

6. Por último, las pruebas "uno-a-uno" pueden ayudar al arquitecto de información a investigar casos específicos más profundamente. Por ejemplo, una búsqueda de información bien planificada (y orquestada) es una buena prueba de cuán intuitiva es la organización del sitio.

Otro método interesante es la clasificación de tarjetas (vea la sección de Recursos más adelante en este artículo). Esta técnica requiere que los participantes junten conceptos parecidos (cada uno en una tarjeta individual). El resultado parece un esquema de organización crudo. En conjunto, el grupo de estos resultados pueden conformar las bases de un prototipo de la arquitectura de información del sitio.

El arquitecto de información debe evitar introducir prejuicios durante las pruebas "uno-a-uno". No debe discutir sus acciones ni tampoco debe corregir u ofrecer asistencia a los participantes; de hecho, debe seguir un guión. Sin embargo, se le recomienda alentar a los interesados a expresar sus opiniones claramente mientras hacen la prueba; esta información puede dar un buen vistazo de cómo navegan su propio espacio conceptual tras bastidores.

Obviamente, el trabajo del arquitecto de información no consiste únicamente en obtener información; debe también analizar esta información, sintetizarla, y probablemente descartar algunos datos o enfocarse en ciertas partes específicas. Entonces, debe hacer algunas buenas conjeturas al azar -- éste no es un campo guiado por es-

trictos lineamientos científicos. Jesse James Garrett, un prominente arquitecto de información, dice que la clave de su éxito es adivinar correctamente.

La arquitectura de información es un proceso, no un resultado final. Muchos sitios han sido lanzados con arquitecturas preliminares, estructuras y organizaciones que son “suficientemente buenas” por el momento. Aún si el arquitecto fuese a desarrollar la organización perfecta para un sitio, la misma no duraría seis meses, debido a que en ese tiempo seguramente algo va a cambiar (las necesidades de los usuarios, objetivos del negocio, etc.) Estos cambios debilitarían la arquitectura de información.

La mejor herramienta que trae a su labor el arquitecto de información es una mente abierta. El arquitecto de información no trata con catálogos de partes fáciles de organizar. Por lo contrario, el campo comprende algo mucho más abstracto -- espacios conceptuales -- y ésto requiere conocer psicología, los fundamentos de la comunicación, y el lenguaje (en particular, la semántica y la pragmática). Además, debido a que el proceso se orienta hacia el contenido, son un factor las abstracciones de la búsqueda, creación, edición, producción, y mantenimiento del contenido.

Dado que el arquitecto de información es el amo del dominio anteriormente mencionado, su responsabilidad es proveer conocimiento -- ser un “líder intelectual”. Sus decisiones y recomendaciones no solo impactan el desarrollo técnico del sitio, sino también las prácticas del negocio. Si los usuarios no pueden encontrar el contenido que buscan, o si el sitio los confunde, no van a regresar. Éstas fallas podrían afectar los ingresos de la compañía y su prestigio en el mercado.

Software y herramientas

Existen muchas aplicaciones de software que prometen suplir mucho del trabajo que hacen los arquitectos de información -- eso es, comprender el contenido y cómo es categorizado, publicado, y administrado. Muchas de éstas herramientas son muy primitivas aun, y no hacen todo lo que prometen. Éstas faltas se deben principalmente al tipo del campos que abarca la arquitectura de información: los lenguajes humanos (y los espacios conceptuales de los seres humanos) son difíciles de analizar, y aún más difíciles de sistematizar en procesos repetibles.

Las herramientas de administración de contenido intentan facilitar el trabajo de la administración del contenido. En esencia, permiten a los equipos de creadores del contenido publicarlo y administrarlo, ya sea en archivos planos, tablas de bases de datos, o ambos.

Desafortunadamente, muchas de éstas herramientas están diseñadas desde un punto de vista superficial del proceso que atraviesa cada unidad de contenido durante toda su existencia. Ésto puede ser debido a que los diseñadores de dichos sistemas tienden a no pedir una descripción de su trabajo a los profesionales de contenido.

La mayoría de los sistemas de administración de contenido le permiten crear una unidad de contenido y procesarla mediante un flujo de trabajo. Dependiendo del sistema, los flujos de trabajo pueden ser fáciles o difíciles de configurar y ajustar a las necesidades de los flujos departamentales establecidos en la organización. Aún los mejores sistemas de administración de contenido tienden a ignorar asuntos vitales para la creación del contenido, como lo son la investigación, la creación de bosquejos, el almacenaje de materiales originales, etc. o inclusive el manejo eficiente del contenido después de su publicación.

Sin embargo, dada la cantidad de proveedores de sistemas de administración de contenido que hay en el mercado, puede ser que la competencia eventualmente produzca una herramienta que valga la pena. Hasta ahora, mi experiencia sugiere que las herramientas basadas en software libre (open source) entalladas para las necesidades de un grupo de personas específicos son mejores que las herramientas comerciales en todas las instancias.

Las herramientas de categorización automática pretenden aliviar la onerosa tarea de categorizar (asignar metadata a) grandes cantidades de contenido, como los repositorios de archivos, tablas de bases de datos, e inclusive archivos de correo electrónico.

Algunas de éstas herramientas no ofrecen una forma fácil de manejar archivos de multimedia, imágenes, e inclusive documentos complejos como los flujogramas de Visio. Además, todavía no entienden significados -- por ejemplo, no distinguen entre el sustantivo amo y la conjugación amo del verbo amar, ni pueden clasificar documentos diferentes que tratan de Java (la isla), Java (el café), y Java (el lenguaje de programación).

Las mejores herramientas de categorización automática permiten a los operadores humanos intervenir de forma juiciosa en el proceso. Por ejemplo, yo ayudé a diseñar una herramienta que ofrece una lista de categorías disponibles para ser aplicadas a una unidad de contenido en base a las sugerencias del operador humano. En éste sistema, es responsabilidad del operador humano escoger las categorías más apropiadas de la lista pre-filtrada que ofrece el sistema, y en caso de que no existan, escoger categorías de la lista completa.

Otro enfoque ha sido crear herramientas que implementan y mantienen diccionarios ideológicos. Un diccionario ideológico es una herramienta que emula las referencias de escritorio usadas por los escritores. Para cada término disponible en una lista de términos, el diccionario puede hacer evidentes las relaciones jerárquicas, y señalar términos alternos (sinónimos) y términos relacionados (conceptos aledaños al significado del término). Un diccionario ideológico bien construido puede aliviar mucha de la frustración que sufren los usuarios al buscar información.

Por ejemplo, si se está desarrollando un sitio de comercio electrónico que se especializa en la venta de ropa, un diccionario ideológico puede evitar la frustración de los usuarios al buscar "blazers", cuando el sistema les llama "chaquetas". Con un diccionario ideológico, es posible definir el término "blazer" como sinónimo de "chaqueta" y desplegar los términos correctos en los resultados de la búsqueda. De la misma manera, cuando el usuario está viendo una chaqueta específica, una columna en la página puede mostrar pantalones que complementan la chaqueta debido a que hay una relación definida entre ambos artículos en el diccionario ideológico.

Muchos de éstos diccionarios son efectivos debido a que combinan lo mejor de ambos mundos. Al operador humano se le permite hacer lo que mejor hace: utilizar su cerebro poderoso y su conocimiento tácito sobre un campo particular. Al computador también se le permite a hacer lo que mejor hace: utilizar sus vastos recursos computacionales para el procesamiento rápido de grandes cantidades de data.

Recursos

- La lista de recursos de arquitectura de información de Jesse James Garrett es un excelente sitio para comenzar.
- El artículo de Jesse James Garrett sobre el estado actual de la arquitectura de información es inteligente e interesante. Es lectura requerida para cualquiera que esté contemplando un cambio de carrera.
- El "Glosario de Arquitectura de Información" de Kat Hagedorn es una lista útil de términos importantes utilizados en el campo, definidos y entrelazados.
- El artículo "Evaluando la Arquitectura de Información" de Steve Toub presenta las posibles consecuencias de no tener una organización efectiva en un sitio. También incluye algunos métodos para estructurar el contenido.
- Keith Instone administra un sitio útil llamado "Usable Web". Está lleno de temas sobre usabilidad y arquitectura de información, e incluye vínculos a recursos en el Web.
- El Centro Argus para la Arquitectura de Información era un sitio muy conocido hasta que fue cerrado hace un año. Sin embargo, todavía contiene muchos artículos útiles y un glosario de términos.
- El sitio de Benjamin Fry contiene artículos interesantes sobre la visualización de data. Su tesis de MIT sobre el diseño orgánico de información es esencial para cualquiera que desee hacer trabajo serio en éste campo.

- El libro "Arquitectura de Información Para el World Wide Web" de Louis Rosenfeld y Peter Morville es considerado un clásico en éste campo.

- La "Clasificación de tarjetas y análisis de grupos" (developerWorks, Enero 2002) de Thomas Myer ofrece una introducción completa a ésta técnica para obtener información sobre los usuarios.

- El artículo "Usabilidad de contenido web" (developerWorks, Mayo 2002) de Thomas Myer presenta guías para hacer el contenido más usable.

- Lea "La experiencia del usuario" (developerWorks, Octubre 2000) de Dick Berry para ver por qué son importantes los modelos de usuarios.

Acerca del autor: Thomas Myer ha sido escritor, desarrollador de multimedia, desarrollador Web, y arquitecto de información. Ha escrito y diseñado proyectos interactivos para compañías como Cisco Systems y Vignette, y actualmente es un consultor independiente y escritor.

Original en Español: Aclaración de Ideas Erróneas Sobre la Arquitectura de Información. Traducido por Jorge Arango

Original en Inglés: Information Architecture Concepts - Misconceptions Explained

Un vocabulario visual para describir arquitectura de información y diseño de interacción

Jesse James Garrett

Tabla de Contenidos

1. Resumen
2. Historia de versiones
3. Consideraciones iniciales
4. Tránsito conceptual
5. Elementos simples: páginas, documentos y pilas de éstos
6. Creando relaciones: conectores y flechas
7. Todo de una vez: conjuntos concurrentes
8. Separándolo: puntos de continuación
9. Elementos comunes: áreas y áreas iterativas
10. Componentes re-utilizables: áreas de flujo y referencias
11. Conceptos básicos para elementos condicionales
12. Haciendo elecciones: puntos de decisión
13. Buscando camino: conectores y flechas condicionales
14. Elección múltiple: ramas condicionales
15. Elige uno o más: selectores condicionales
16. Una decisión, muchos caminos: racimos
17. Algunas restricciones pueden aplicar: áreas condicionales
18. Conclusión
19. Librerías descargables de formas

Resumen

Los diagramas son una herramienta esencial para comunicar arquitectura de información y diseño de interacción en equipos de desarrollo Web. Este documento trata las consideraciones en el desarrollo de tales diagramas, delinea una simbología básica para diagramar conceptos de arquitectura de información y diseño de interacción, y entrega guías para el uso de estos elementos.

Historia de versiones

1.1b (6 Mar 2002)

Información sobre soporte incorporado en OmniGraffle 2.0

Nueva librería de formas para iGrafX Flowcharter

1.1a (17 Sep 2001)

Nueva librerías de formas para Macromedia FreeHand

Publicada hoja de trucos y template de formas PDF

1.1 (31 Ene 2001)

Agregado el elemento pila de archivos

Agregado el elemento selector condicional

Modificado el elemento flecha para permitir múltiples puntas de flecha

Modificado el comportamiento del elemento racimo de tal forma que ahora sólo aparece flujo abajo desde un selector o rama condicional

Modificado el comportamiento del elemento rama condicional para permitir un resultado nulo

Numerosas mejoras a las librerías de formas

Nueva librería de formas para Adobe InDesign

1.0 (17 Oct 2000)

Publicación inicial

Consideraciones iniciales

Un vocabulario visual es un conjunto de símbolos usado para describir algo (usualmente un sistema, estructura o proceso). El vocabulario descrito aquí puede ser usado por un arquitecto de información o diseñador de interacción para describir, en un nivel alto, la estructura y/o flujo de la experiencia de usuario de un sitio Web.

Estas descripciones, o diagramas son usados por cinco audiencias primarias:

- Inversionistas y gerentes de proyecto, los utilizan para obtener un sentido general del alcance y forma del proyecto.
- Productores de Contenido, los usan para derivar los requerimientos de contenido.
- Diseñadores visuales y de interfaces, los utilizan para derivar una cuenta de cuántos diseños de pagina únicos deben ser producidos y obtener un sentido inicial de los requerimientos de navegación e interfaz para estos diseños.
- Tecnólogos, los utilizan para derivar requerimientos funcionales.
- Arquitectos de información y diseñadores de interfaz los usan para desarrollar requerimientos detallados de navegación e interfaz para cada página.

Cada una de estas audiencias (exceptuando los inversionistas) necesita gran cantidad de detalle para hacer su trabajo. El problema es que el detalle que cada audiencia requiere varía en gran manera del detalle requerido por los otros, y la mayoría de este detalle es irrelevante para las necesidades de otras audiencias.

El enfoque sensible es limitar el detalle en el diagrama a lo que puede ser útilmente aplicado por todas las audiencias. El diagrama por lo tanto sirve como un documento hito para el desarrollo de documentos más detallados, específicos a las necesidades de cada audiencia.

Otros requerimientos clave de un vocabulario visual para arquitectura de información y diseño de interacción incluyen:

- Compatible con pizarra blanca: El vocabulario debería ser tan simple que los diagramas puedan ser dibujados rápidamente a mano. Los elementos del vocabulario debieran ser suficientemente distintos entre sí para que un dibujo medianamente malo no comprometa la claridad del diagrama.
- Independiente de herramienta: El vocabulario debiera estar diseñado de forma que no requiera de software especializado para construir diagramas. El vocabulario no debiera

favorecer el uso de una herramienta particular de software, sino permitir a los arquitectos utilizar las herramientas más cómodas a ellos.

- Pequeño y auto-contenido: Porque estos diagramas son usados por una amplia gama de audiencias con diferentes niveles de conocimiento (o incluso interés) en sistemas de diagramas usados en otras áreas de desarrollo técnico, el vocabulario no debiera requerir tal conocimiento o interés. El total de los elementos debe ser mantenido al mínimo posible, manteniendo una estricta relación uno-a-uno entre conceptos y símbolos, para que el vocabulario pueda ser aprendido y aplicado en forma rápida. Los conceptos expresados por el diagrama pueden ser arbitrariamente complejos; el medio de su expresión no debe serlo.

Trasfondo conceptual

Arquitectura de información y diseño de interacción son dos caras de la misma moneda. (Ver "Los Elementos de la Experiencia de Usuario" para definiciones de los términos como son usados aquí.) Los diagramas de sitios contemporáneos inevitablemente involucran ambas caras. Pero para cada una, los objetivos del diagrama son levemente diferentes.

En ambos casos, el diagrama se enfoca en lo que llamamos la macro-estructura, entregando sólo detalle suficiente para permitir a los miembros del equipo ver la "gran foto". La tarea del arquitecto es determinar el nivel apropiado de detalle para lograr este objetivo. El detalle específico a nivel de página o micro-estructura, es detallado en otros documentos de los cuales el arquitecto puede no ser directamente responsable de desarrollar.

Cuando describimos arquitectura de información, el diagrama debiera enfatizar la estructura conceptual y organización del contenido. Nótese que la estructura conceptual no es lo mismo que la organización de navegación. El objetivo del diagrama de arquitectura de información no es entregar una especificación de navegación completa; este detalle es mejor puesto en otros documentos, donde cauce menos riesgo de confundir y distraer.

Cuando describimos diseño de interacción, el diagrama debiera enfatizar cómo el usuario fluye a través de tareas definidas, y lo que son los pasos discretos en estas tareas. Tal como con la navegación, los detalles de interfaz no debieran aparecer en el diagrama - si te encuentras dibujando botones y campos, probablemente estás cargando el diagrama con un exceso de detalle.

Este vocabulario está basado en un modelo conceptual simple abarcando tanto arquitectura de información como diseño de interacción:

- Al sistema presenta al usuario caminos.
- El usuario se mueve a través de estos caminos mediante acciones.
- Estas acciones entonces causan al sistema a generar resultados.

Elementos simples: paginas, documentos y pilas de éstos

La unidad básica de la experiencia de usuario en la Web es por supuesto, la pagina, la cual representamos con un simple rectángulo. Nótese que la pagina es una unidad de presentación, no (necesariamente) una unidad de implementación -- una pagina en tu diagrama puede representar múltiples archivos HTML (como en una interfaz con frames) o unidades múltiples de código (como en includes en el servidor -- SSI -- o una implementación manejada por bases de datos).

Además de paginas, también hay archivos, parcelas de datos sin propiedades de navegación. Estos archivos son entregados al usuario para su uso fuera de un ambiente de navegador Web, ó browser (tales como archivos de video, archivos independientes como PDFs ó ejecutables). Para estos, usamos nuestro viejo amigo el icono con oreja de perro.



Figura 1a: [izquierda] La pagina y el documento
 Figura 1b: [derecha] La pila de paginas y la pila de documentos

Usa una pila de paginas para indicar un grupo de paginas funcionalmente idénticas cuyas propiedades de navegación son inmateriales a la macro-estructura del sitio. Similarmente, una pila de documentos representa un grupo de documentos que reciben tratamiento de navegación idéntico y pueden ser clasificadas como una entidad única (tal como una colección de juegos descargables o una librería de manuales de instrucciones en PDF).

Usamos etiquetas en paginas y archivos para identificarlos. Éstas no tienen la necesidad de ser una correlación con designaciones como el elemento <TITLE> HTML o nombres de documentos, pero deben ser únicos para cada página o documento en el diagrama. Identificadores numéricos únicos y designaciones de tipo también entregan una buena forma de llevar el rastro todas las paginas y documentos en tu diagrama.

Creando relaciones: conectores y flechas

Las relaciones entre los elementos son ilustradas mediante líneas simples o conectores. Estas relaciones conceptuales se traducirán inevitablemente en relaciones de navegación -- pero no todas las relaciones de navegación aparecerán en el diagrama.

En el caso de la arquitectura de información, éstas relaciones están comúnmente reflejadas a través de la organización jerárquica de paginas en árboles. Sin embargo, esto de ninguna manera es obligatorio ni (en algunos casos) recomendable.

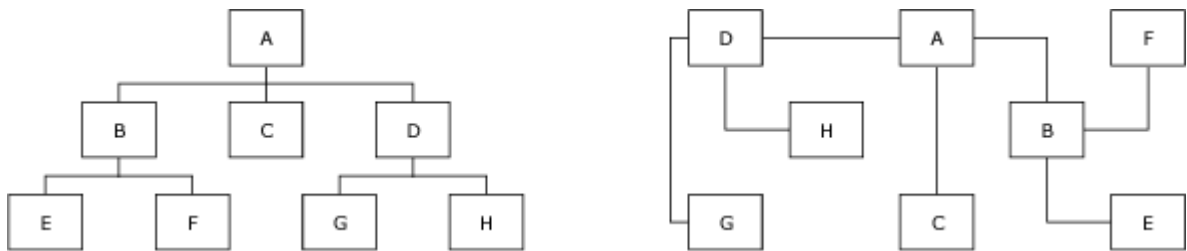


Figura 2a: [izquierda] Una estructura simple de árbol
 Figura 2b: [derecha] La misma estructura diagramada de forma diferente

Cuando diagramamos diseño de interacción, nuestras líneas también deben indicar dirección para indicar cómo el usuario se moverá a través del sistema por una tarea particular. Transformando nuestros conectores en flechas hará el truco de forma limpia. Usamos los términos corriente abajo y corriente arriba para referirnos a la posición de los elementos relativa a este movimiento hacia adelante.

Nótese que estas flechas no son como las flechas que indican una calle de un solo sentido, más bien son como las flechas que indican el camino hacia el patio de comida en el centro comercial. El usuario no está impedido de moverse en dirección opuesta; la flecha indica solamente la dirección en la cual el usuario probablemente querrá ir.

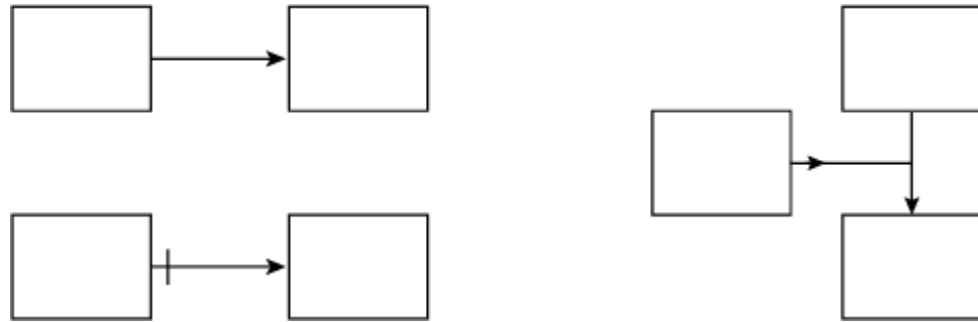


Figura 3a: [arriba izquierda] Flecha indica movimiento corriente abajo hacia el fin de la tarea
 Figura 3b: [abajo izquierda] Barra cruzada indica que el movimiento corriente arriba no está permitido
 Figura 3c: [derecha] Flechas múltiples clarifican la dirección

Si por alguna razón queremos prohibir el movimiento corriente arriba (como en los casos donde alguna acción irreversible como eliminar un registro ha tomado lugar), usamos una barra cruzada (sólo una corta línea perpendicular) en el extremo opuesto a la punta de la flecha para indicar esto.

En algunos casos, puede ser necesario agregar una punta de flecha adicional cerca de la página corriente arriba para clarificar la dirección del flujo en una arquitectura más compleja. (Una nota práctica: muchas aplicaciones de diagramación no permiten al usuario encadenar dos flechas de esta manera. Para solucionar esto, las librerías de formas incluyen un elemento "punto de goma", un elemento invisible que consiste en un punto único de conexión. Usa este elemento para unir dos flechas.)

Los conectores y flechas también pueden ser etiquetados, pero el uso de éstas debe ser limitado a casos en los cuales la acción tomada por el usuario necesita ser clarificada. Si las etiquetas se tornan largas, son muchas y comienzan a sobrecargar el diagrama, apunta al lector hacia una nota al pie o una referencia en un anexo.

En los ejemplos dados en este documento, referencias al pie de página y en anexos aparecerán como una combinación de número y letra entre paréntesis. Los números se refieren a la página de diagrama en la cual la referencia aparece; las letras se refieren a la nota específica. Por ejemplo, la primera nota en la página 3 de un diagrama debería ser referida como (3a), la segunda (3b) y así en adelante.

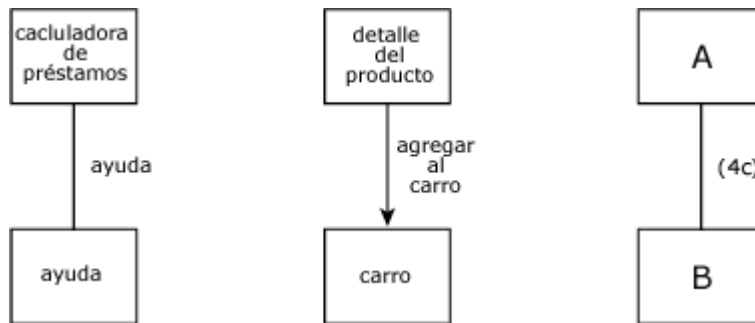


Figura 4a: [izquierda] Una etiqueta superflua
 Figura 4b: [centro] Una etiqueta útil
 Figura 4c: [derecha] Una referencia al pie o anexo

Todo de una vez: conjuntos concurrentes

Un conjunto concurrente (representado por el semi-círculo) es usado en casos cuando una acción del usuario genera resultados múltiples simultáneos (tal como abrir una ventana pop-up mientras una página se carga en la ventana principal, o mostrar una pagina mientras un documento es descargado).



Figura 5: Un conjunto concurrente

Como las flechas, los conjuntos concurrentes tienen dirección. Elementos corriente arriba se conectan al lado curvo; elementos corriente abajo se conectan al lado plano.

Separándolo: puntos de continuación

Los arquitectos de información se encuentran a menudo deseando hojas de papel más grandes para diagramar su trabajo. Pero aun si dispositivos de salida de gran formato como plotters fueran ampliamente disponibles, algunas arquitecturas son simplemente muy complejas para ser capturadas en un diagrama único que lo incluya todo.

Para permitirnos separar nuestros diagramas en secciones fáciles de digerir, usamos puntos de continuación (paréntesis cuadrado) para unir los vacíos entre las páginas.

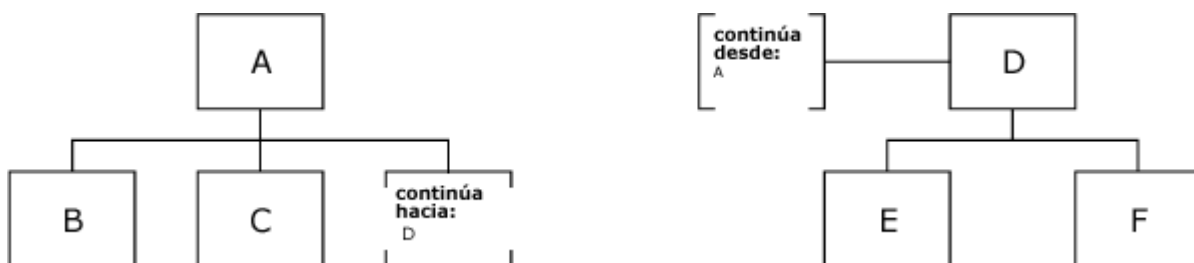


Figura 6a: [izquierda] Un punto "continúa hacia" referencia al lector hacia otro diagrama
 Figura 6b: [derecha] Un punto "continúa desde", retomando desde donde salimos de 6a

Un punto de continuación único puede listar una o más fuentes o destinos según se necesite. La orientación de los corchetes (horizontal y vertical) no tiene significado particular; la elección de orientación es problema del juicio estético del arquitecto.

Elementos comunes: áreas y áreas iterativas

El elemento área (un rectángulo de esquinas redondeadas) es usado para identificar un grupo de paginas que comparten uno o más atributos comunes (tales como aparecer en una ventana pop-up, o tener un tratamiento único de diseño). Usa etiquetas para identificar estos atributos o (como con los conectores), haz referencias a notas fuera del documento si tienes mucho que decir.

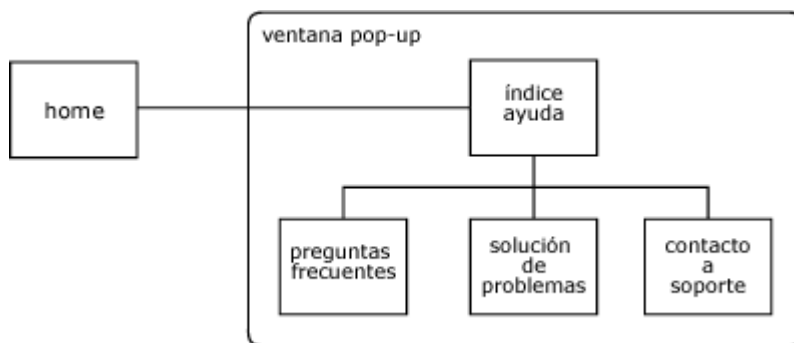


Figura 7: Un ejemplo de uso de un área para representar una ventana pop-up

Muchas arquitecturas incluyen repetir la misma estructura básica tal como es aplicada a un número de elementos de información funcionalmente idénticos. Por ejemplo, puedes tener un catálogo de productos en el cual cada producto tiene varias páginas asociadas a él. Podrías dibujar una instancia de esta estructura para cada producto, pero, ¿por qué perder el tiempo? Simplemente usa un área iterativa -- una pila de rectángulos con esquinas redondeadas -- en vez.

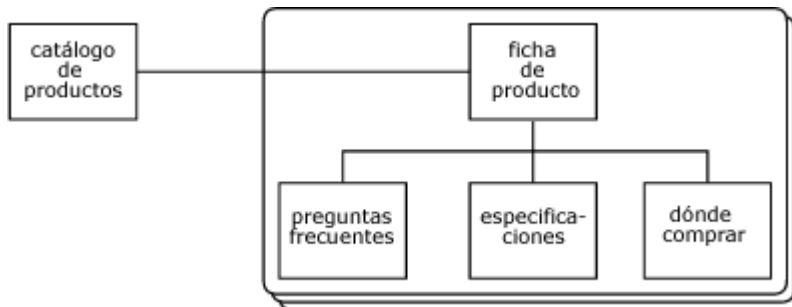


Figura 8: Un ejemplo de uso de un área iterativa para representar una estructura repetida en un catálogo de productos

Nótese que conectores y flechas no apuntan a las áreas mismas. Los elementos de área sólo sirven para encerrar las paginas. Las áreas deben ser aplicadas con cuidado, es muy fácil pillarse capturando toda clase de detalles con elementos de área que no se manifiestan en la experiencia de usuario (tal como qué páginas son alojadas en cuáles servidores) o de otra manera interfieren con el objetivo general del diagrama de comunicar la macro-estructura.

Componentes re-utilizables: áreas de flujo y referencias

Algunos diseños de interacción requieren que una secuencia de pasos (como por ejemplo, un procedimiento de login) aparezca repetidamente en diferentes contextos a través del diseño. A menudo estas secuencias son meramente un componente de una o más tareas que el usuario está tratando de lograr. (Esto es análogo al concepto de sub-rutina en programación de ordenadores).

Tal área re-utilizable es llamada un flujo, y es representada en el diagrama mediante dos elementos: el área de flujo, que encierra el flujo mismo; y la referencia de flujo, que sirve como marcador para el flujo en cada contexto en el cual se repite. Ambos elementos tienen la misma forma básica, un rectángulo con las esquinas cortadas (o, si prefieres, un octágono desfigurado).

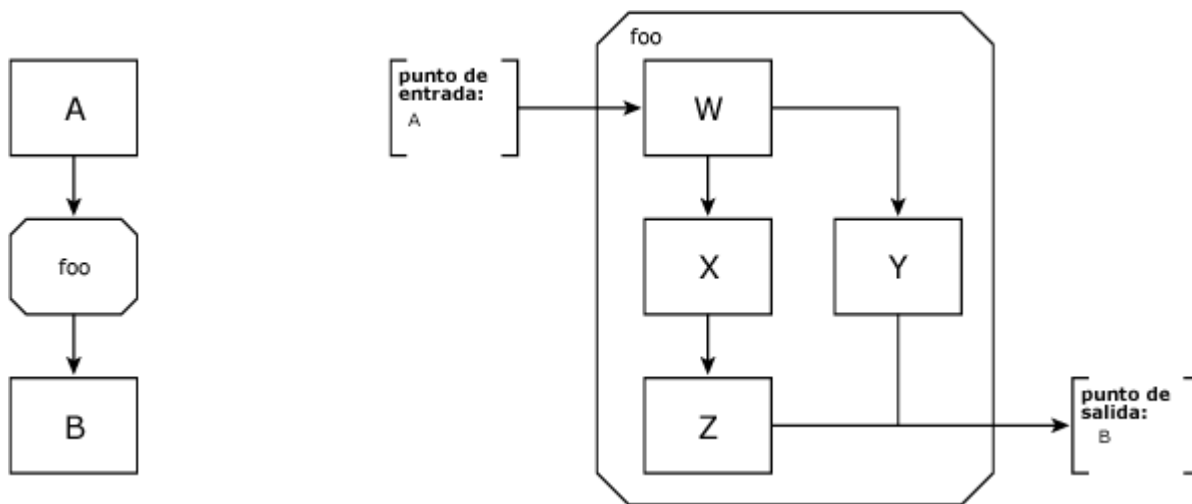


Figura 9a: [izquierda] Una referencia de flujo sirve tanto como punto "continúa hasta", como punto "continúa desde"

Figura 9b: [derecha] El área de flujo referida en 9a

Las áreas de flujo también requieren el uso de dos tipos de puntos de continuación especiales: puntos de entrada y puntos de salida. Éstos son ubicados fuera del área de flujo, mientras los

puntos de continuación, dentro del área de flujo, indican que el flujo abarca múltiples diagramas.

Las referencias de flujo en sí mismas funcionan de manera muy similar a los puntos de continuación. El objetivo de ambos tipos de elementos es el mismo: permitir al arquitecto cortar el diagrama en páginas. La diferencia es que una referencia de flujo puede ser usada en ambas modalidades; "continúa desde" y "continúa hasta", mientras que un punto de continuación puede sólo ser uno u otro. Si no necesitas un elemento que tenga ambos roles, probablemente no necesitas usar un flujo.

Conceptos básicos para elementos condicionales

Con cada vez mayor frecuencia, las arquitecturas de información y diseños de interacción son reformados de manera dinámica por el sistema mientras el usuario se mueve a través del sitio. Esta reformación es lograda mediante lógica condicional, y los elementos restantes de este vocabulario son específicos a estructuras de lógica condicional. He aquí un modelo conceptual básico para la aplicación de elementos condicionales:

- El sistema sigue la pista a uno o más atributos, estos atributos pueden ser particulares a:
 - o el usuario (tal como el tipo de usuario)
 - o la sesión (tal como el estado del login)
 - o el contenido siendo accedido (tal como materia temática)
 - o o pueden existir "en el mundo" (tal como la hora y fecha).
- Los atributos tienen valores ("3 p.m." es un valor posible para "hora del día").
- La asociación de un atributo con un valor particular es llamada una condición.
- Las condiciones son evaluadas por el sistema para determinar si son verdaderas.

En una arquitectura estática, cada camino es presentado a todos los usuarios bajo toda circunstancia, y cada camino conduce al mismo resultado. En una estructura dinámica, el sistema decide cuáles caminos o resultados son presentados al usuario basado en la evaluación de una o más condiciones.

Para minimizar la sobrecarga en nuestros diagramas, estas condiciones son típicamente descritas en una nota al pie o anexo que acompaña al diagrama.

Haciendo elecciones: puntos de decisión

Cuando una acción de un usuario puede generar uno de un número de resultados, el sistema debe tomar una decisión acerca de cuál resultado debe presentar. (Probablemente el ejemplo más común de esto es manejo de errores en el envío de formularios.) Llamamos a esto un punto de decisión, y como en diagramas de flujo tradicionales, es representado por un diamante.

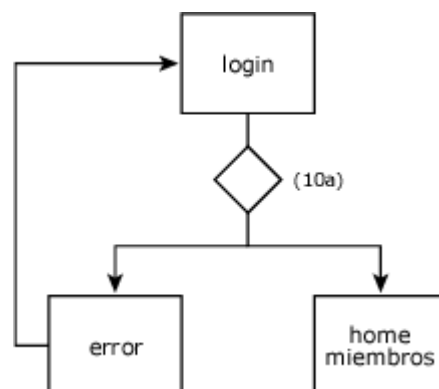


Figura 10: Un ejemplo de uso de un punto de decisión en una secuencia de login

Nótese que las fechas deben ser usadas en conjunto con los puntos de decisión, para clarificar si los elementos asociados se encuentra corriente arriba o corriente abajo desde el punto de decisión.

Buscando camino: conectores y flechas condicionales

Un conector condicional (representado por una línea cortada) es usado cuando un camino puede ser o no ser presentado al usuario dependiendo de si una o más condiciones son cumplidas.



Figura 11a: [izquierda] Un conector condicional
 Figura 11b: [derecha] Una flecha condicional

Por ejemplo, puede haber una página que contenga información sensible que sólo puede ser vista por empleados de la compañía. La condición en este caso sería el tipo de usuario (empleado); si la condición se cumple, el camino se hace disponible. Si no, no existe camino.

Elección múltiple: ramas condicionales

Cuando un sistema debe seleccionar un camino entre un numero de opciones mutuamente exclusivas a ser presentadas al usuario, usamos una rama condicional (triángulo). Los elementos corriente arriba se conectan a un punto del triángulo; los elementos corriente abajo se conectan al lado opuesto.

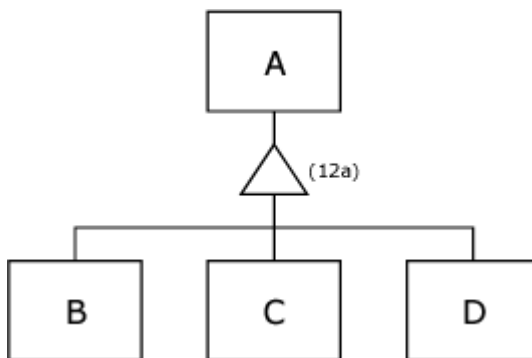


Figura 12: Una rama condicional

El ejemplo mostrado en la figura 12 se ve muy parecido al ejemplo del punto de decisión arriba en la figura 10, pero el comportamiento descrito aquí es bastante diferente. En el ejemplo del punto de decisión, sólo un camino (o elemento de navegación) era presentado al usuario; dónde conducía al usuario ese elemento dependía de ciertas condiciones.

En la figura 12, el sistema está tomando una decisión similar, pero sucede antes que la acción del usuario. La rama condicional indica que el sistema está decidiendo cuál camino será presentado al usuario. Los caminos desde la pagina A hacia las páginas B, C y D son mutuamente exclusivos; por ejemplo si existe un camino hacia B, los caminos hacia C y D no existen.

Tal como con los conectores y flechas condicionales, una rama condicional puede entregar al usuario ningún camino (un resultado nulo). La diferencia aquí es que con una rama condicional un resultado nulo está prohibido; y de estar permitido, es uno de tres o más resultados posibles. Indica si una rama permite un resultado nulo en una nota al pie de página o indicación en un anexo.

Elige uno o más: selectores condicionales

El elemento selector condicional (representado por un trapezoide) funciona de manera muy similar a la rama condicional, con una diferencia importante: con el selector, los varios caminos corriente abajo no son mutuamente exclusivos, cualquier número de caminos que satisfagan las condiciones pueden ser presentados al usuario.

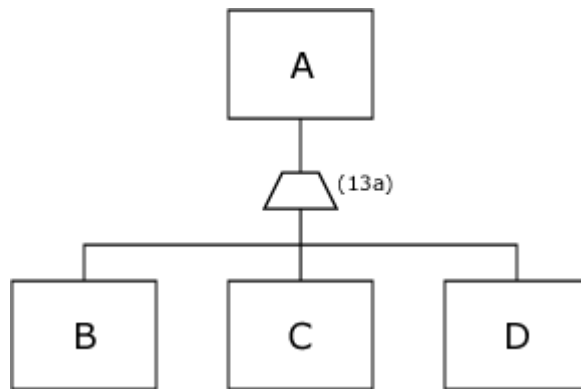


Figura 13: Un selector condicional

La aplicación más común del selector condicional es en resultados generados por un motor de búsqueda. En este caso, la pagina de resultados de búsqueda aparecería corriente abajo desde el selector; la condición es el criterio de búsqueda ingresado por el usuario; los caminos corriente abajo llevarían a las paginas de contenido indexadas por el motor de búsqueda. Tal como con una rama condicional, el selector condicional puede generar un resultado nulo -- de hecho, es mucho más común con un selector que con una rama.

Una decisión, muchos caminos: racimos

Algunas estructuras condicionales requieren que el sistema presente más de un camino basado en ciertas condiciones. Asociamos estos caminos en la estructura con un racimo (representado por un círculo). El racimo puede aparecer corriente abajo desde una rama condicional o un selector condicional.

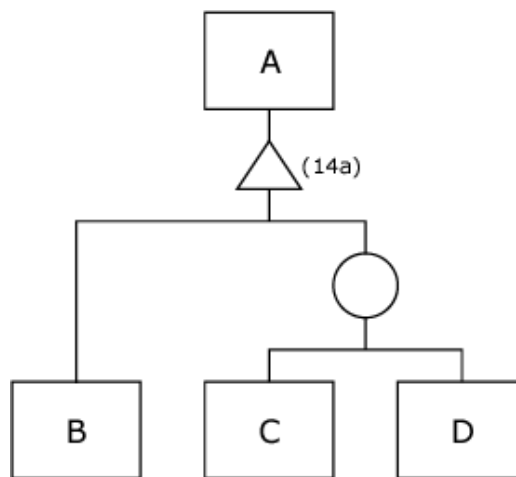


Figura 14: Un racimo corriente abajo desde una rama

La estructura dibujada en la Figura 14 funciona de forma muy similar a una rama condicional, pero por una condición estamos presentando más de un camino al usuario. Entonces, si el atributo siendo evaluado tiene valor x, el usuario ve un camino hacia la pagina B; pero si el atributo tiene valor y, el usuario ve caminos hacia ambas paginas C y D.

Algunas restricciones pueden aplicar: áreas condicionales

Cuando una o más condiciones aplican a un grupo de páginas, esas páginas son encerradas en un área condicional -- un rectángulo de esquinas redondeadas, pero con un tratamiento de línea cortada como el conector condicional.

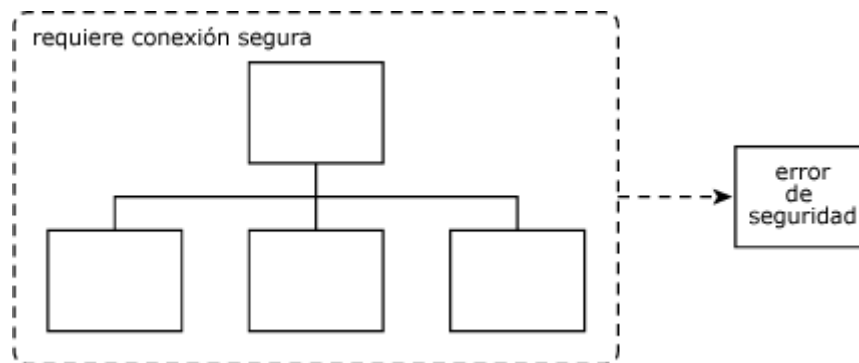


Figura 15: Un ejemplo de uso de un área condicional donde se requiere una conexión segura

Las áreas condicionales se usan comúnmente en situaciones que involucran permisos de acceso, como cuando se requiere un login o conexión encriptada (SSL). A diferencia de otros tipos de áreas, las áreas condicionales son asociadas con un resultado, el cual se genera en caso de que la(s) condición(es) no son satisfechas.

Conclusión

Si deseas ver cómo se arma el sistema completo, aquí hay un diagrama de muestra de la arquitectura de información y diseño de interacción de MetaFilter. (No estuve involucrado en el desarrollo de este sitio; este diagrama fue desarrollado a partir del sitio.)

Scott Larson creó esta práctica hoja de trucos para una referencia rápida a los varios elementos condicionales. Y para aquellos interesados en crear sus propias librerías de formas para usar en una aplicación diferente a las que aparecen abajo, aquí hay un PDF de todas las formas (gracias a Ross Olson por la sugerencia).

El vocabulario necesariamente representa sólo un primer paso. A medida que la arquitectura de información y diseño de interacción continúan su evolución, aparecerán situaciones que este vocabulario no abarca. Tu retroalimentación y recomendaciones para la próxima versión de este vocabulario son bienvenidas.

Librerías descargables de formas

OmniGraffle 2.0 es la primera aplicación en despacharse con soporte incorporado para el vocabulario visual. OmniGraffle viene actualmente pre-instalado en todos los Apple Power Mac y PowerBook. También puede ser descargado desde el sitio Omni.

Otras librerías de formas disponibles:

- Archivo de patrón para Visio 2000
- Archivo de patrón para Visio 5
- Archivo de patrón para Visio 4
- Archivo PowerPoint
- Archivo de librería para Adobe InDesign
- Archivo de librería para FreeHand 10 (gracias Andrew Crow)
- Archivo de librería para FreeHand 9 (gracias Andrew Crow)
- Archivo EPS Illustrator
- Archivo de librería para iGrafX Flowcharter 2000 (gracias Andrew Robinson)
- Colección de archivos SVG (1.9 MB)
- Colección de archivos EPS que contienen un elemento por archivo, útil para ser importado en otras aplicaciones (1.1 MB)