Information architecture

DEFINITION, DESIGN FOR FINDING & UNDERSTANDING, VISUALIZATION

POLITECNICO DI TORINO

Laura Farinetti - DAUIN

e-Lite

User-centered design

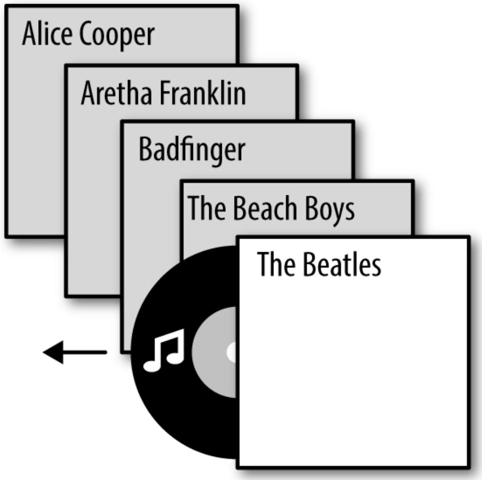
- Definition
- Human Computer Interaction
- User experience
- Usability and accessibility
- The design process
- Information architecture
- Evaluation

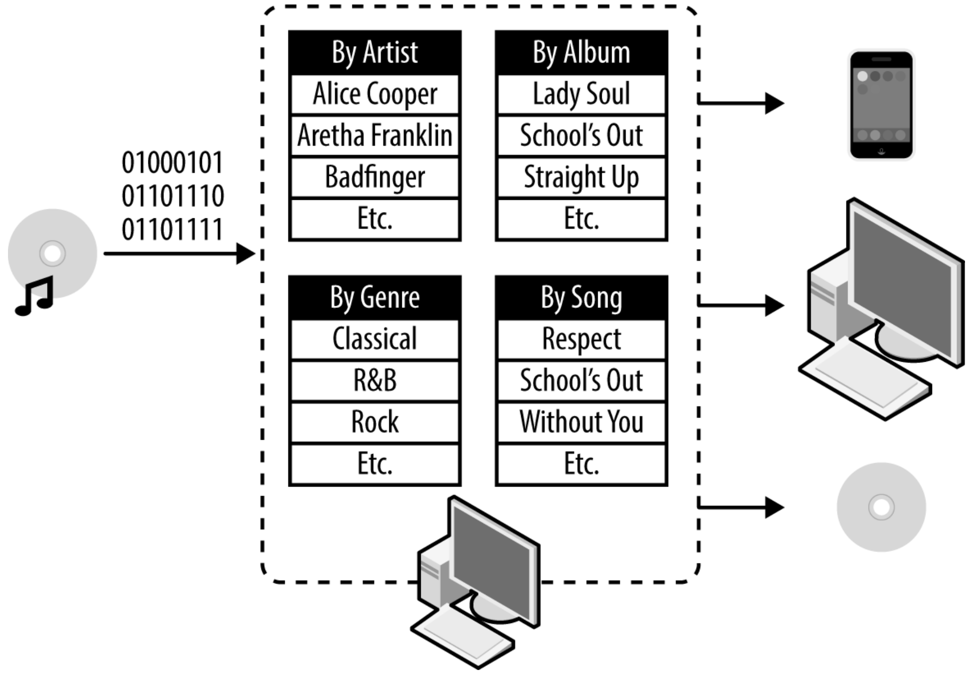
Information architecture

- Information architecture (IA) is a design discipline focused on making information findable and understandable
- IA allows to think about problems through two important perspectives:
  - That information products and services are perceived by people as places made of information
  - That these information environments can be organized for optimum findability and understandability

Information

- Information embedded in physical objects
  - Need to choose how to organize on the shelf





- Dematerialized information
  - Can be organized in more than one way

IA definition(s)

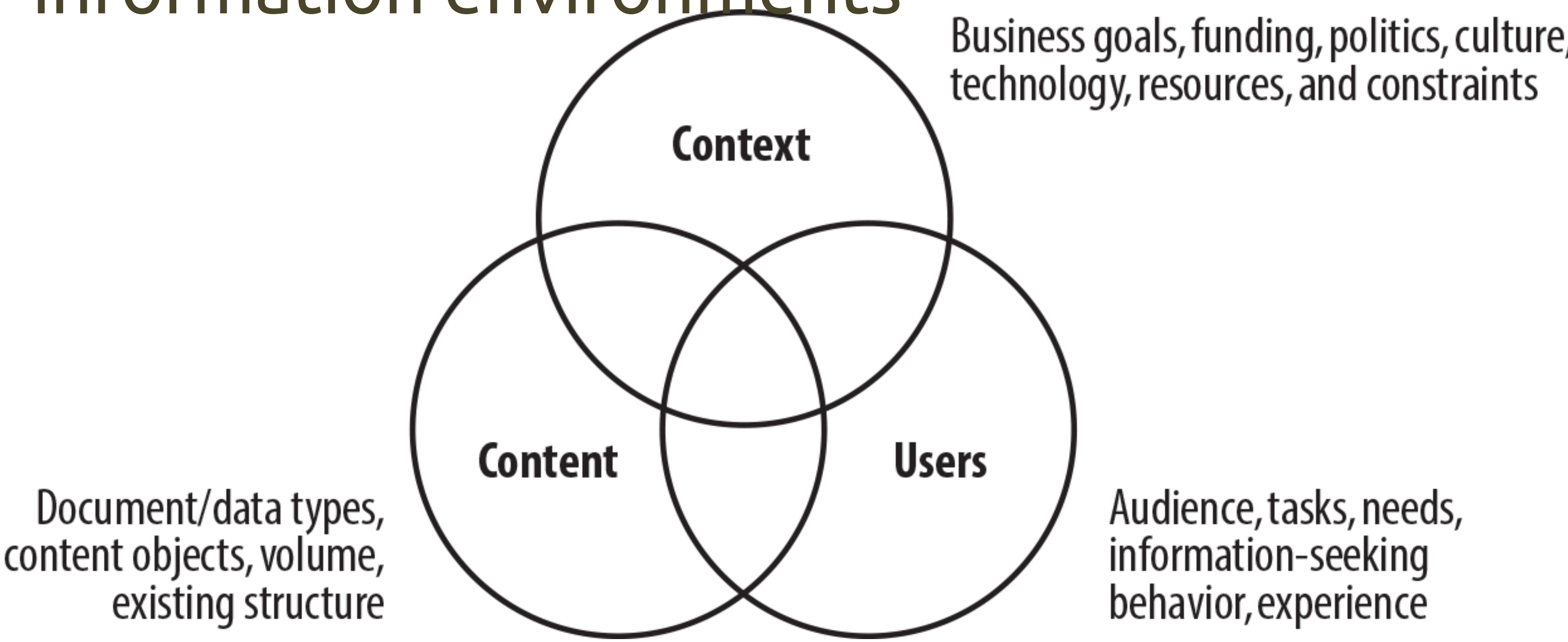
1. The structural design of shared information environments
2. The synthesis of organization, labeling, search, and navigation systems within digital, physical, and cross-channel ecosystems
3. The art and science of shaping information products and experiences to support usability, findability, and understanding
4. An emerging discipline and community of practice focused on bringing principles of design and architecture to the digital landscape

Information

- Information architecture is not data and knowledge management
- Data is facts and figures
  - Relational databases are highly structured and produce specific answers to specific questions
- Knowledge is the stuff in people's heads
  - Knowledge managers develop tools, processes, and incentives to encourage people to share that stuff
- Information is in the middle
  - With information systems, there's often no single "right" answer to a given question
  - Information can be of all shapes and sizes: websites, documents, software applications, images, and more
  - Metadata: terms used to describe and represent content objects such as documents, people, processes, and organizations

Model for effective information architecture

- Concept of “information ecology” to address the complex dependencies that exist in information environments



Context

- All digital design projects exist within a particular business or organizational context
  - Whether explicit or implicit, each organization has a mission, goals, strategy, staff, processes and procedures, physical and technology infrastructure, budget, and culture
- The key to success is understanding and alignment
  - First, you need to understand the business context
  - What makes it unique? Where is the business today, and where does it want to be tomorrow? In many cases, you're dealing with tacit knowledge
  - Then, you need to find ways to align the information architecture with the goals, strategy, and culture of the business

Content

- “Content” are the documents, applications, services, schemas, and metadata that people need to use or find in the systems
- Many facets
  - Ownership: who creates and owns the content?
  - Format: databases, reports in word or PDF, videos, …
  - Structure: some information systems are built around the document paradigm, with the fully integrated document as the smallest discrete unit; other systems allow management and access at a finer level of granularity
  - Metadata: to what extent has metadata that describes the content and objects within your system already been created? Have documents been tagged manually or automatically? What’s the level of quality and consistency
  - Volume: how much content are we talking about? A hundred applications? A thousand pages? A million documents? How big is the system?
  - Dynamism: what is the rate of growth or turnover? How much new content will be added next year? And how quickly will it go stale?

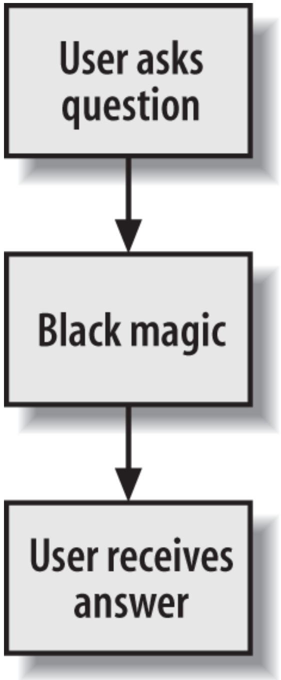
Users

- Users are human beings with desires, needs, concerns, and foibles
- Do you know who's using your system? Do you know how they're using it? And perhaps most importantly, do you know what information they want from your systems?
  - Senior executives may need to find a few good documents on a particular topic very quickly
  - Research analysts may need to find all the relevant documents and may be willing to spend several hours on the hunt
  - Managers may have a high level of industry knowledge but low navigation and searching proficiency
  - Teenagers may be new to the subject area but skilled in handling a search engine

Design for finding & design for understanding

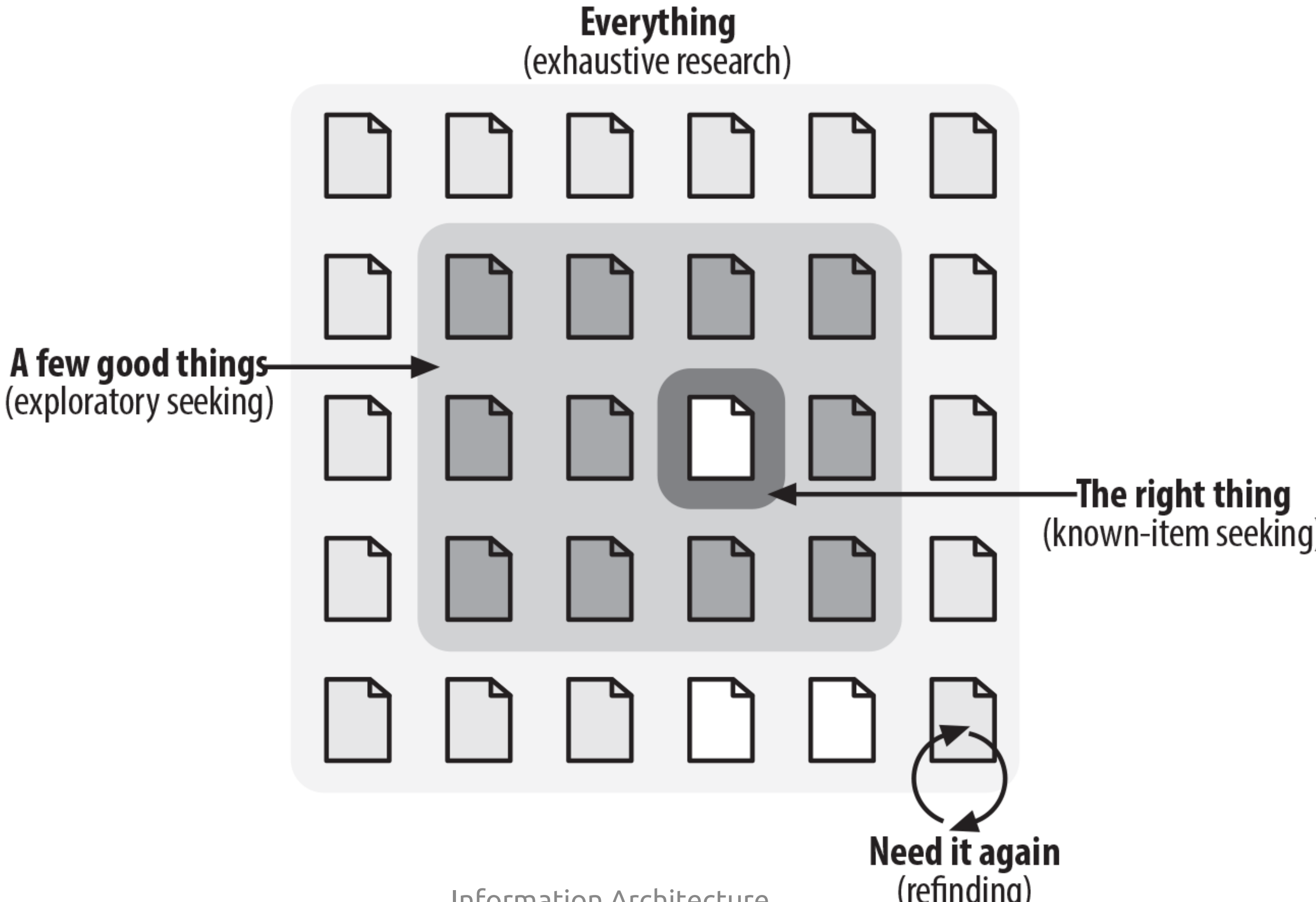
- Information architecture is focused on making information findable and understandable
  - Related, but different objectives

Design for finding



- Information architecture is not restricted to taxonomies or search engines
- It starts with people and the reason they come to your site or use your app: they have an information need
- The "too simple" information model
  - Rarely happens this way
  - Only if users have a question for which there is a right answer, they know where to find the answer, they know how to state the question, and they know how to use the system to do so
- Finding information is not a straightforward problem that can be addressed by a simple, algorithmic approach

Four different types of information needs



Four different types of information needs

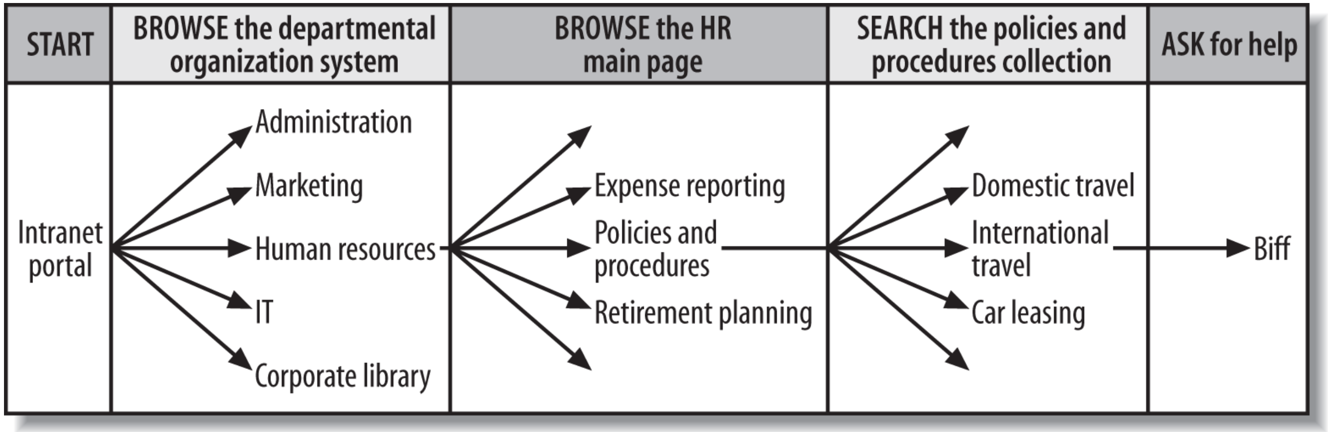
- Known-item seeking
  - You know what you're looking for, what to call it, and where you'll find
  - Example: you search the staff directory to find a colleague's phone number
- Exploratory seeking
  - You're not exactly sure what you're looking for: you're looking to learn something from the process of searching and browsing
  - Example: a user may go to his employer's human resources site to learn something about retirement plans that the company offers
  - In the process, you may encounter some basic information
  - Typically open ended: there is no clear expectation of a "right" answer, the user is happy to retrieve a few good results, and use them as a springboard for the next iteration of the search

Four different types of information needs

- Exhaustive research
  - When you're looking for everything available on a particular topic want everything, hoping to leave no stone unturned
  - In this case, the user often has many ways to express what she's looking for, and may have the patience to construct her search using all those varied terms
  - Example: someone who is trying to learn more about a friend's medical condition
  - Again, there isn't necessarily a "right" answer
- Refinding
  - Our failing memories and busy schedules continually force us to engage in pieces of useful information that we've happened upon before
  - Example: while you're at work, you might surf for a few minutes and stumble on a great but long explanation of something you are interested on, but you won't read it and try to refind it later at home

Design for finding

- How do website users find information? Searching, browsing, and asking (humans)
- Two more aspects in seeking behavior: integration and iteration
  - We often integrate searching, browsing, and asking in the same finding session
  - We don't always get things right the first time



Design for understanding

- We only understand things in relationship to something else
- Information architects are concerned with creating environments that are understandable and usable by human beings, and which can grow and adapt over time to meet the needs of users and their organizations
- We experience information environments as places where we go to transact, learn, and connect with other people, among many other activities
  - When designing information environments, we can learn from the design of physical environments
  - Some organizing principles that carry over to information environments from physical environments include structure and order, rhythm, typologies, and modularity and extensibility

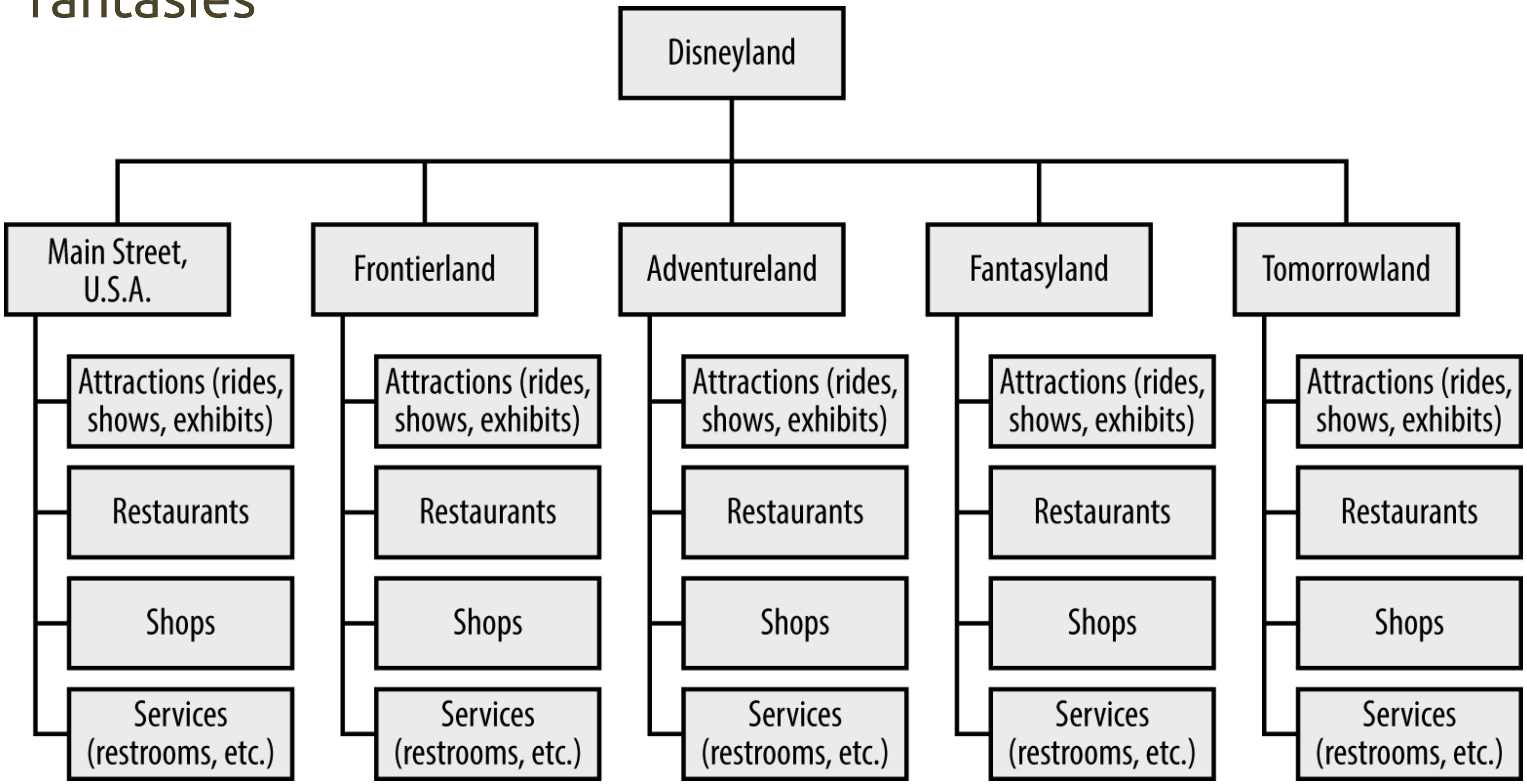
Example: typology

- Most bank (or university, or hospital, or …) buildings are similar

- Most bank (or university, or hospital, or arlines, or online stores, …) websites are similar

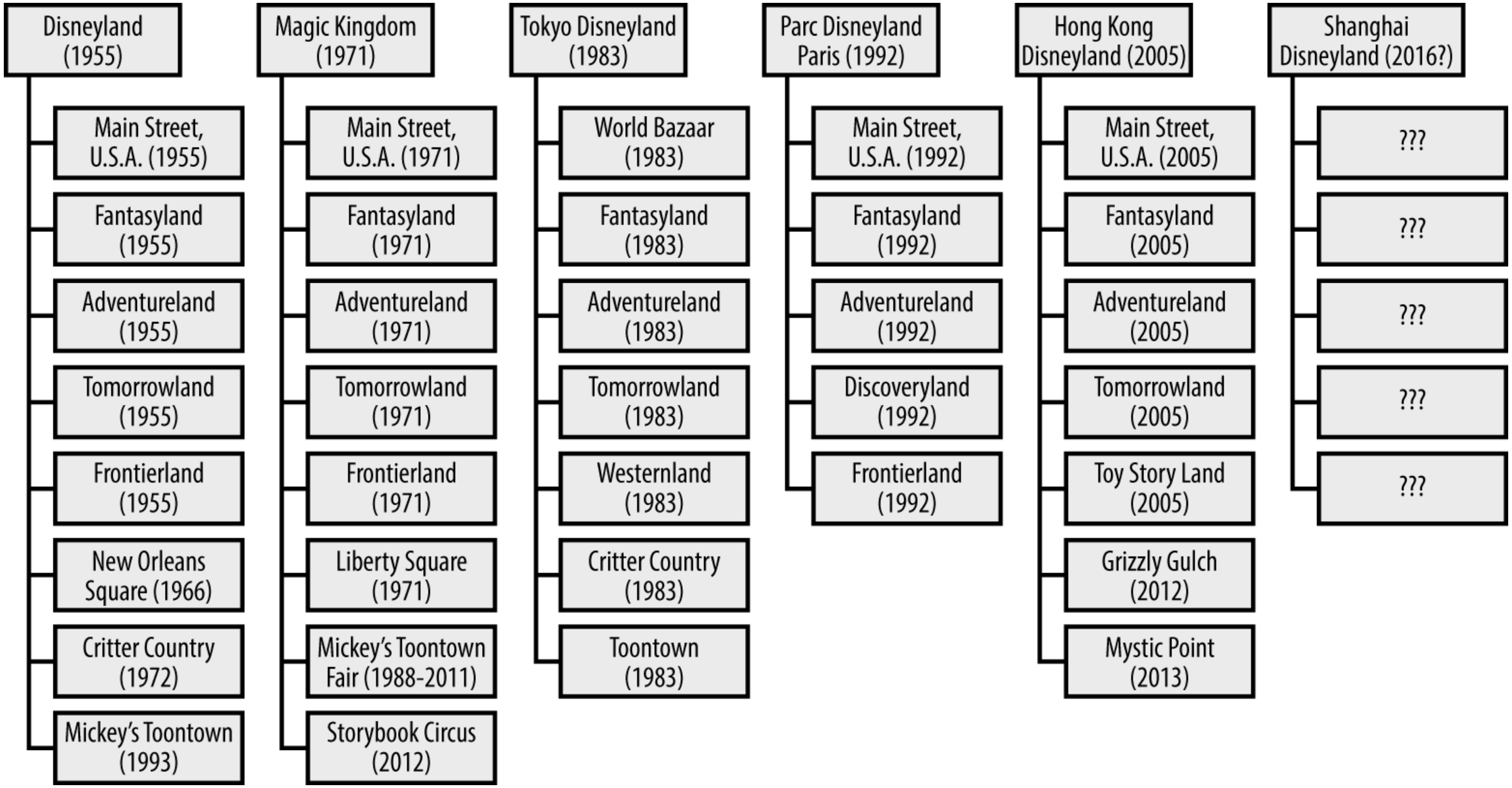
Example: Disneyland

- The organizational structure of Disneyland made a new, unfamiliar concept—the theme park—easily understandable to mid-1950s Americans by appealing to their emotions and fantasies



Example: Disneyland

- Coherence, extensibility, and adaptability to cultural and temporal context



Two different approaches

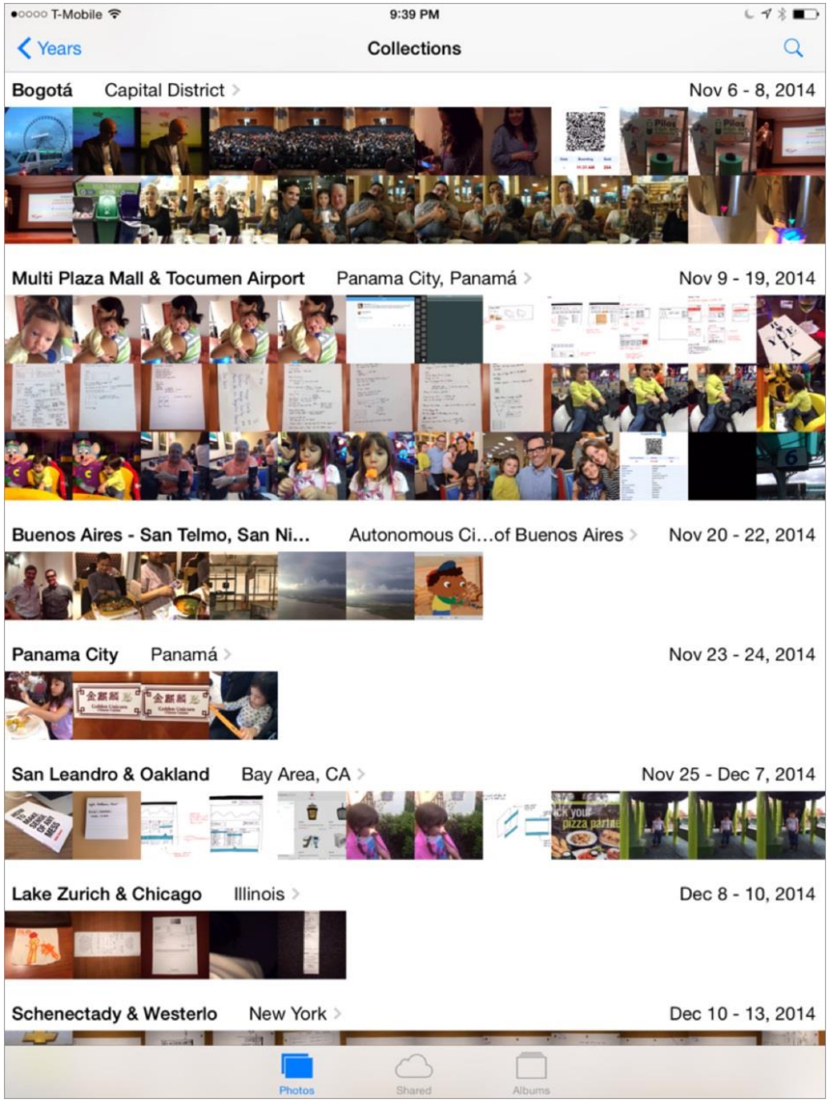
- Top-down information architecture
  - When the site's designers have worked hard to determine the most common questions, and have designed the site to meet those needs
- Bottom-up information architecture
  - As information environments have become more dynamic and search engines have become more powerful and widespread, this modality has gained prominence
  - Instead of being dictated "from above", bottom-up information architecture is suggested by and inherent in the system's content
  - It's important because users are increasingly likely to bypass your system's top-down information architecture

Top-down IA

- Where am I? (1)
- I know what I'm looking for; how do I search for it? (2)
- How do I get around this site? (3)
- What's important and unique about this organization? (4)
- What's available on this site? (5)
- What's happening there? (6)
- How do I engage with them via various other popular digital channels? (7)
- How can I contact a human? (8)
- What's their address? (9)
- How can I access my account? (10)

Bottom-up IA

- There is little to see here besides the information architecture and the content itself
- Information architecture provides context for the content, and tells us what we can do
  - Know where you are
  - Move to other closely related views
  - Move through the information hierarchically and contextually
  - Search the content
  - Share the content



IA components

- Four categories
  - Organization systems: how we categorize information (e.g., by subject or chronology)
  - Labeling systems: how we represent information (e.g., using scientific terminology or everyday terminology)
  - Navigation systems: how we browse or move through information (e.g., clicking through a hierarchy)
  - Searching systems: how we search information (e.g., executing a search query or with an index)

Browsing aids

- These components present users with a predetermined set of paths to help them navigate the information environment
  - When browsing, users don't articulate their queries through search fields, but find their way through menus and links
- Organization systems
  - Also known as taxonomies and hierarchies, these are the main way of categorizing or grouping content (e.g., by topic, by task, by audiences, or by chronology)
  - User-generated tags are also a form of organization system
- General navigation systems
  - Primary navigation systems that help users understand where they are and where they can go within an information environment

Browsing aids

- Local navigation systems
  - Primary navigation systems that help users understand where they are and where they can go within a portion of an information environment (e.g., a subsite)
- Sitemaps/tables of contents
  - Navigation systems that supplement primary navigation systems
  - Provide a condensed overview of and links to major content areas within the environment, usually in outline form
- Indices
  - Supplementary navigation systems that provide an alphabetized list of links to the contents of the environment

Browsing aids

- Guides
  - Supplementary navigation systems that provide specialized information on specific topics, as well as links to related subsets of content
- Walkthroughs and wizards
  - Supplementary navigation systems that lead users through sequential sets of steps
- Contextual navigation systems
  - Consistently presented links to related content
  - Often embedded in text and generally used to connect highly specialized content within an information environment

Search aids

- These components allow the entry of user-defined queries and automatically present users with customized sets of results that match their queries
  - Dynamic and mostly automated counterparts to browsing aids
- Search interface
  - The means of entering and revising a search query, typically with information on how to improve your query, as well as other ways to configure your search (e.g., selecting from specific search zones)
- Query language
  - The grammar of a search query
  - Query languages might include Boolean operators (e.g., AND, OR, NOT), proximity operators (e.g., ADJACENT, NEAR), or ways of specifying which field to search (e.g., AUTHOR="Shakespeare")

Search aids

- Query builders
  - Ways of enhancing a query's performance
  - Common examples include spell checkers, stemming, concept searching, and drawing in synonyms from a thesaurus
- Retrieval algorithms
  - The part of a search engine that determines which content matches a user's query
  - Google's PageRank is perhaps the best known example
- Search zones
  - Subsets of site content that have been separately indexed to support narrower searching (e.g., searching the tech support area within a software vendor's site)
- Search results
  - Presentation of content that matches the user's search query
  - Involves decisions about what types of content should make up each individual result, how many results to display, and how sets of results should be ranked, sorted, and clustered

Content and tasks

- These are the users' ultimate destinations, as opposed to separate components that get users to their destinations
  - However, it's difficult to separate content and tasks from information architecture, as there are components embedded in them that help find the way
- Examples of information architecture components embedded in content and tasks
  - Headings: labels for the content that follows them
  - Embedded links: links within text, that label (i.e., represent) the content they link to
  - Embedded metadata: information that can be used as metadata but must first be extracted (e.g., in a recipe, if an ingredient is mentioned, this information can be indexed to support searching by ingredient)

Content and tasks

- Examples of information architecture components embedded in content and tasks
  - Chunks: logical units of content, that can vary in granularity (e.g., sections and chapters are both chunks) and can be nested (e.g., a section is part of a book)
  - Lists: set of chunks or links to chunks that have been grouped together (e.g., they share some trait in common) and have been presented in a particular order (e.g., chronologically)
  - Sequential aids: clues that suggest where the user is in a process or task, and how far he has to go before completing it (e.g., “step 3 of 8”)
  - Identifiers: clues that suggest where the user is in an information system (e.g., a logo specifying what site she is using, or a breadcrumb explaining where she is)

“Invisible” components

- Certain key architectural components are only in the background: users rarely (if ever) interact with them
  - These components often “feed” other components, such as a thesaurus that’s used to enhance a search query
- Examples of invisible information architecture components
  - Controlled vocabularies and thesauri: predetermined vocabularies of preferred terms that describe a specific domain (e.g., auto racing or orthopedic surgery)
  - Retrieval algorithms: used to rank search results by relevance
  - Best bets: preferred search results that are manually coupled with a search query (e.g., “editor’s choice”)

Documenting IA

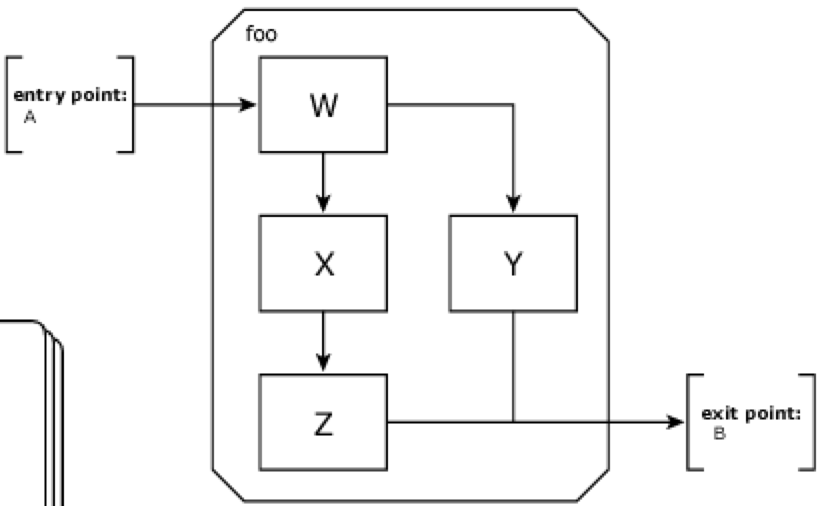
- Provide multiple “views” of an information architecture
  - Information environments are too complex to show all at once: a diagram that tries to be all things to all people is destined to fail
  - No single view takes in the whole picture, but the combination of multiple diagrams might come close
- Develop those views for specific audiences and needs
  - Whenever possible, determine what others need from your diagrams before creating them
  - You might find that a visually stunning diagram is compelling to client prospects, therefore justifying its expense
  - However, it probably requires too many resources to use in a production environment, where diagrams may change multiple times per day

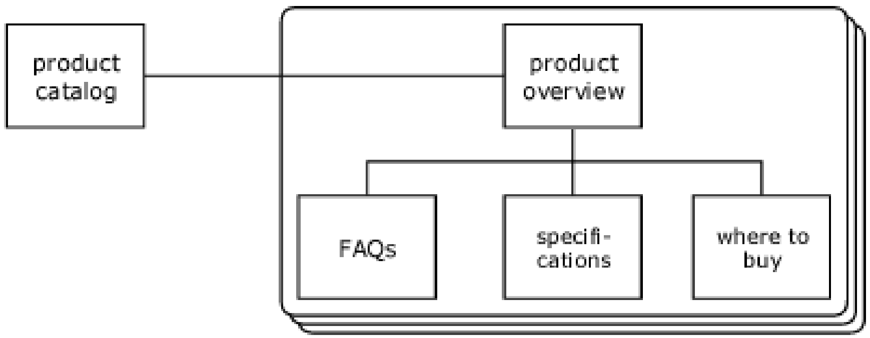
Diagrams

- Diagrams are useful for communicating the two basic aspects of an information system structural elements
  - Content components: what constitutes a unit of content, and how those components should be grouped and sequenced
  - Connections between content components: how content components are linked to enable actions such as navigating between them
- A variety of visual vocabularies have emerged to help convey the complexity of information architecture in visual diagrams
  - Each provides a clear set of terms and syntax to visually communicate components and their links
  - Visual vocabularies are at the heart of the many templates used to develop sitemaps and wireframes

Visual vocabularies

- Example: Jesse James Garrett’s
  - http://www.jjg.net/ia/visvocab/
  - Translated in many languages



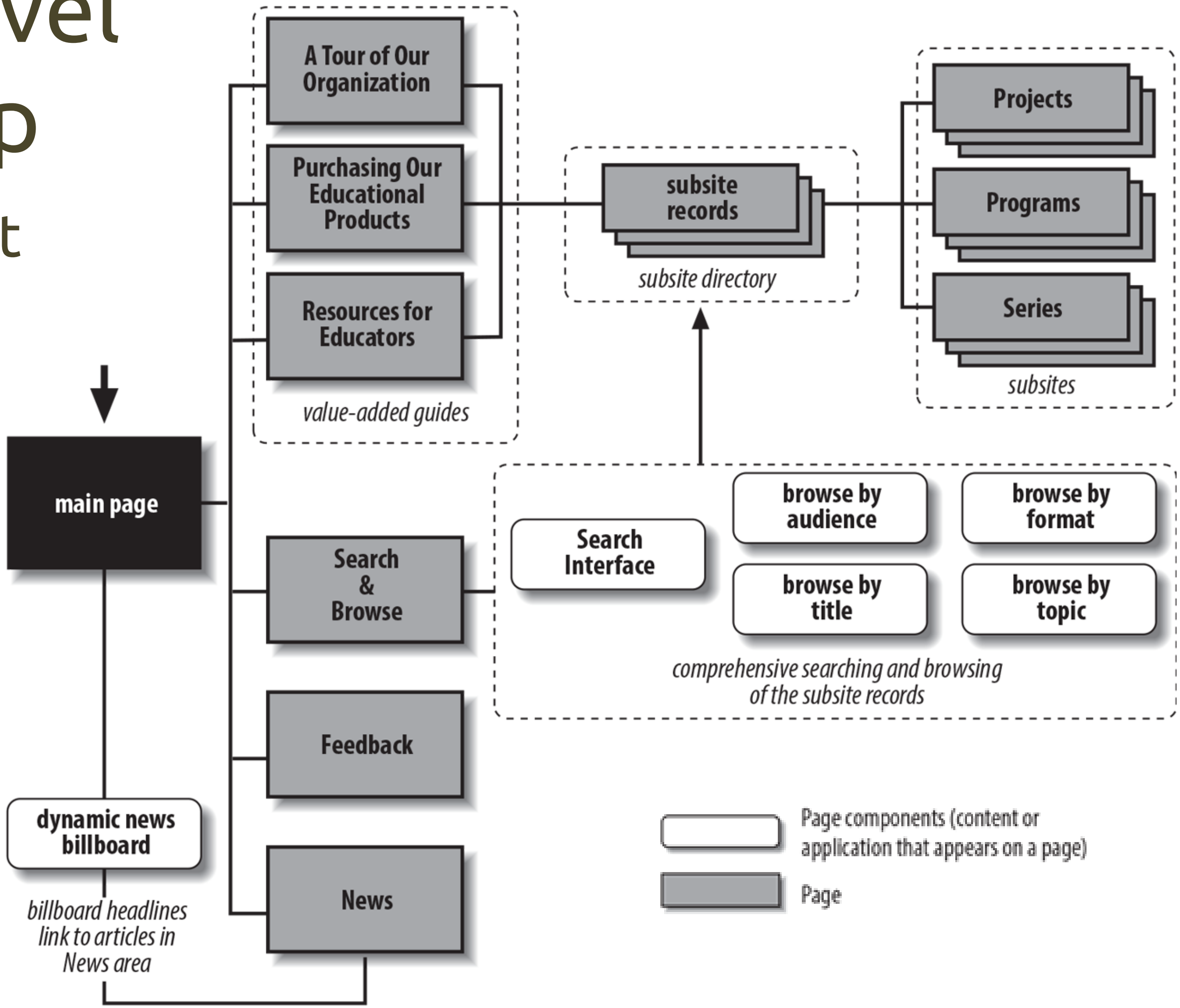


Sitemaps

- Sitemaps show the relationships between information elements such as pages and other content components, and can be used to portray organization, navigation, and labeling systems
- High-Level architecture sitemaps
  - Often created as part of a top-down information architecture process
  - Start with the main page, and iteratively show more and more of the architecture, adding subsidiary sections, increasing levels of detail, and working out the navigation from the top down
  - Most useful for exploring primary organization schemes and approaches: map out the organization and labeling of major areas, usually beginning with a bird's-eye view from the main page of the website

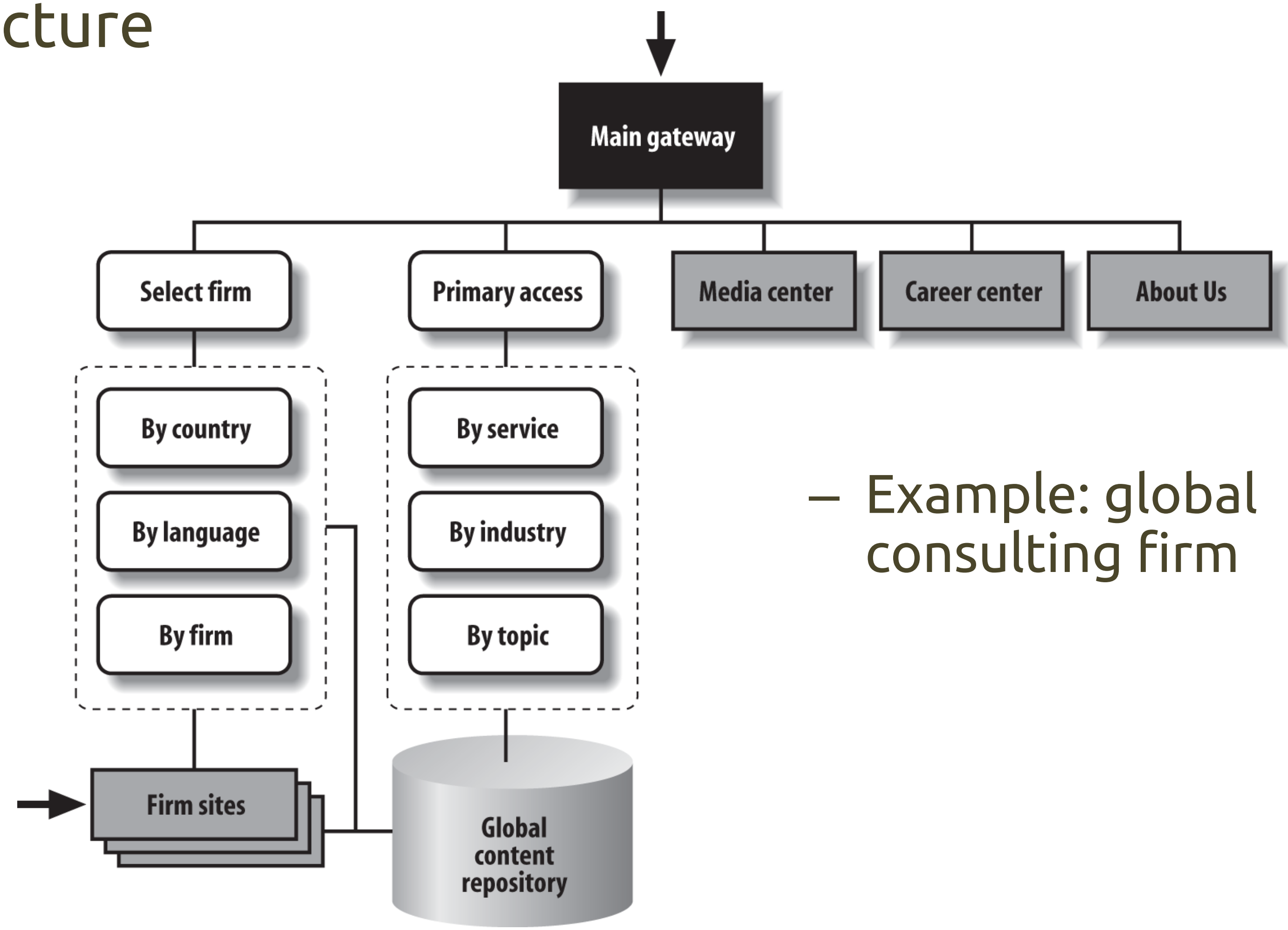
High level sitemap

- Different layouts possible



Sitemaps

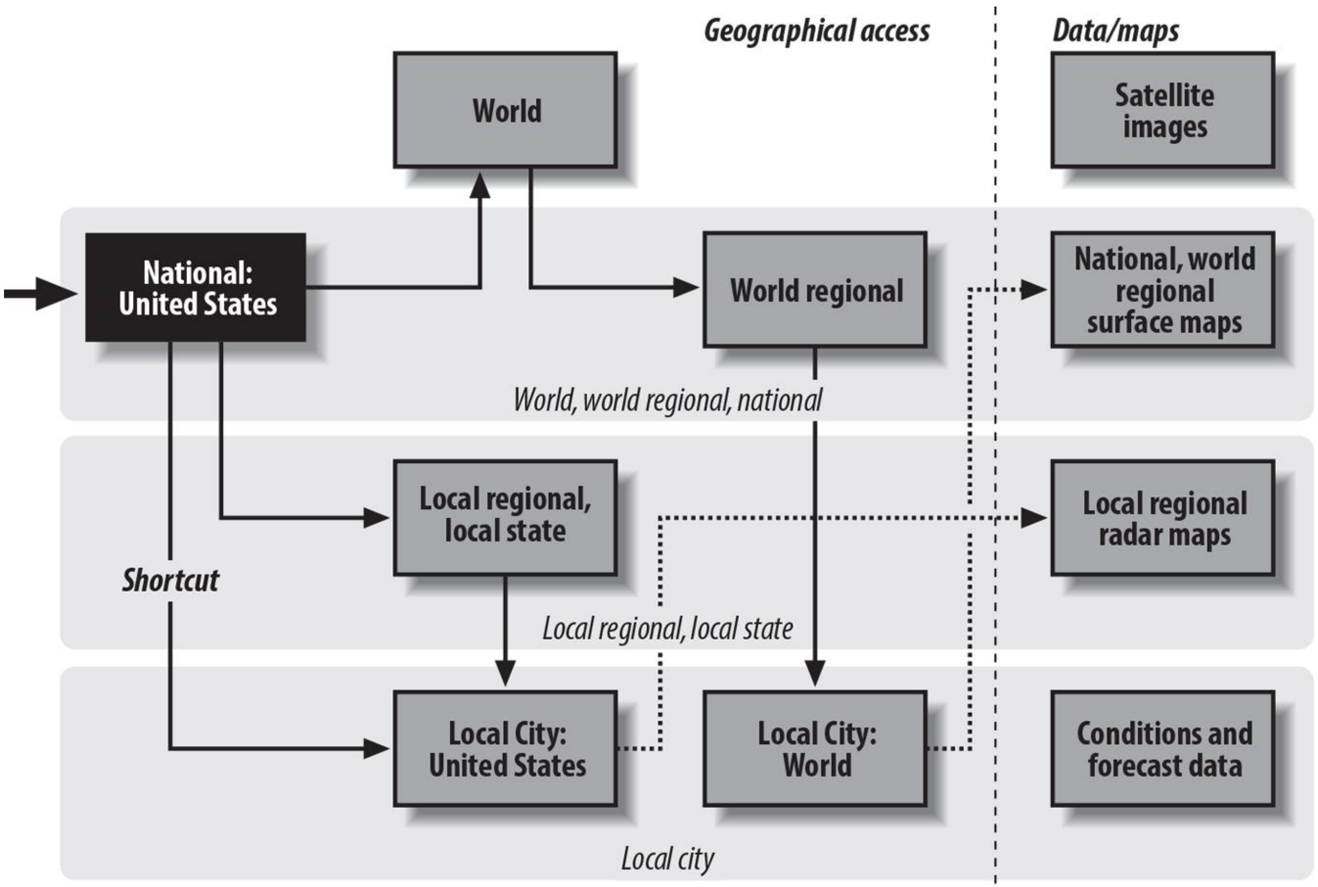
- Can provide a holistic view of the information architecture



  – Example: global consulting firm

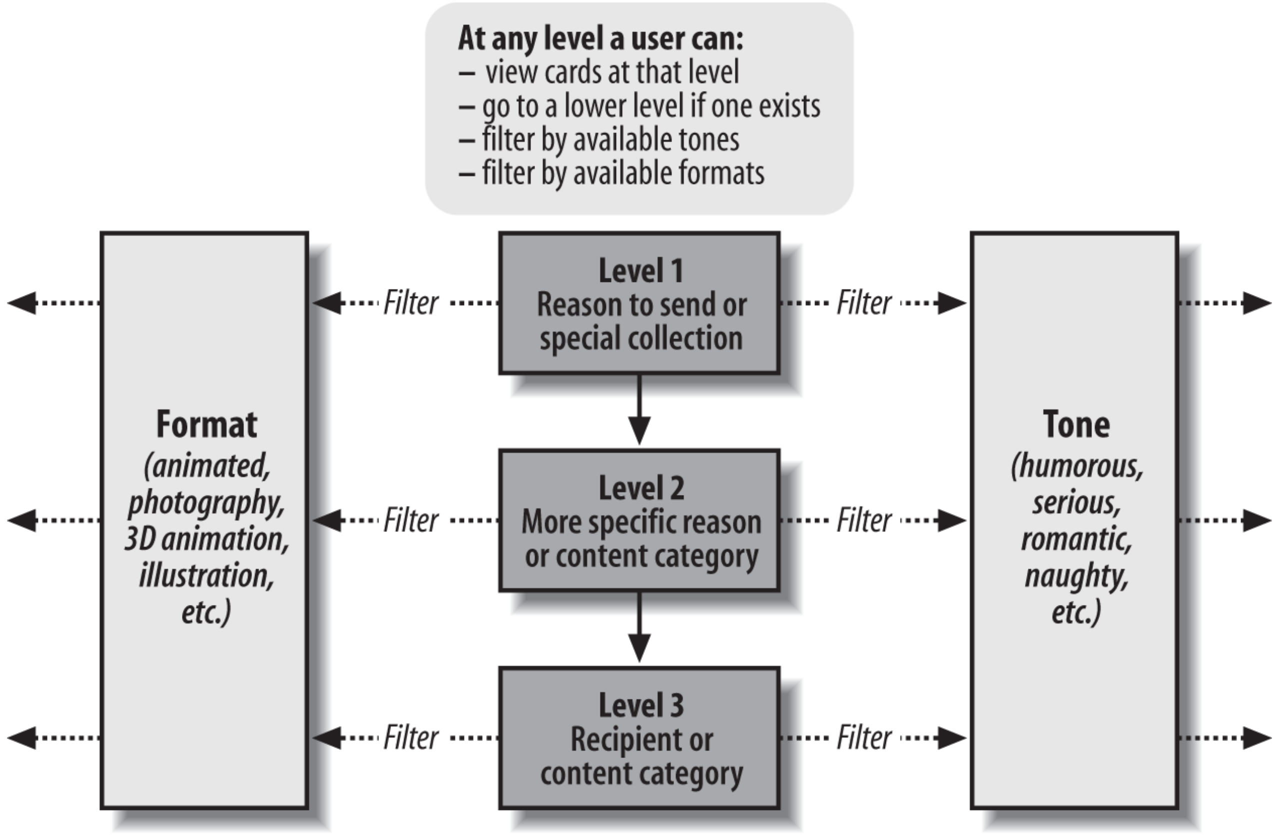
Sitemaps

- Can focus on a single aspect of navigation
  - Example: The Weather Channel's website, how to move between local and national weather reports and news



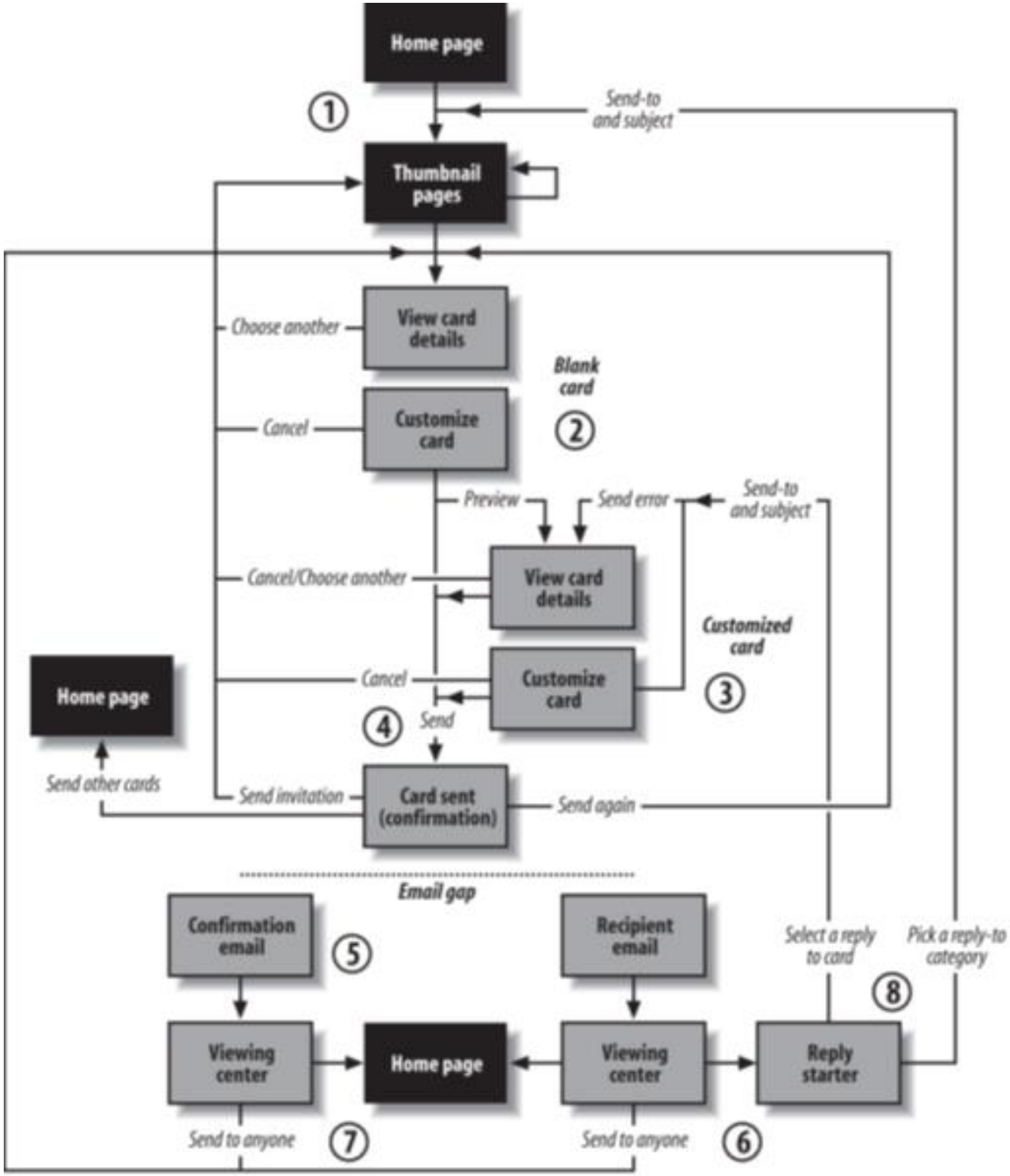
Sitemaps

- Can focus on a specific user's task
  - Example: how filtering might work at Egreetings.com



Sitemaps

- Can present a user-centered view of a process
  - Example: the card-sending process at Egreetings.com

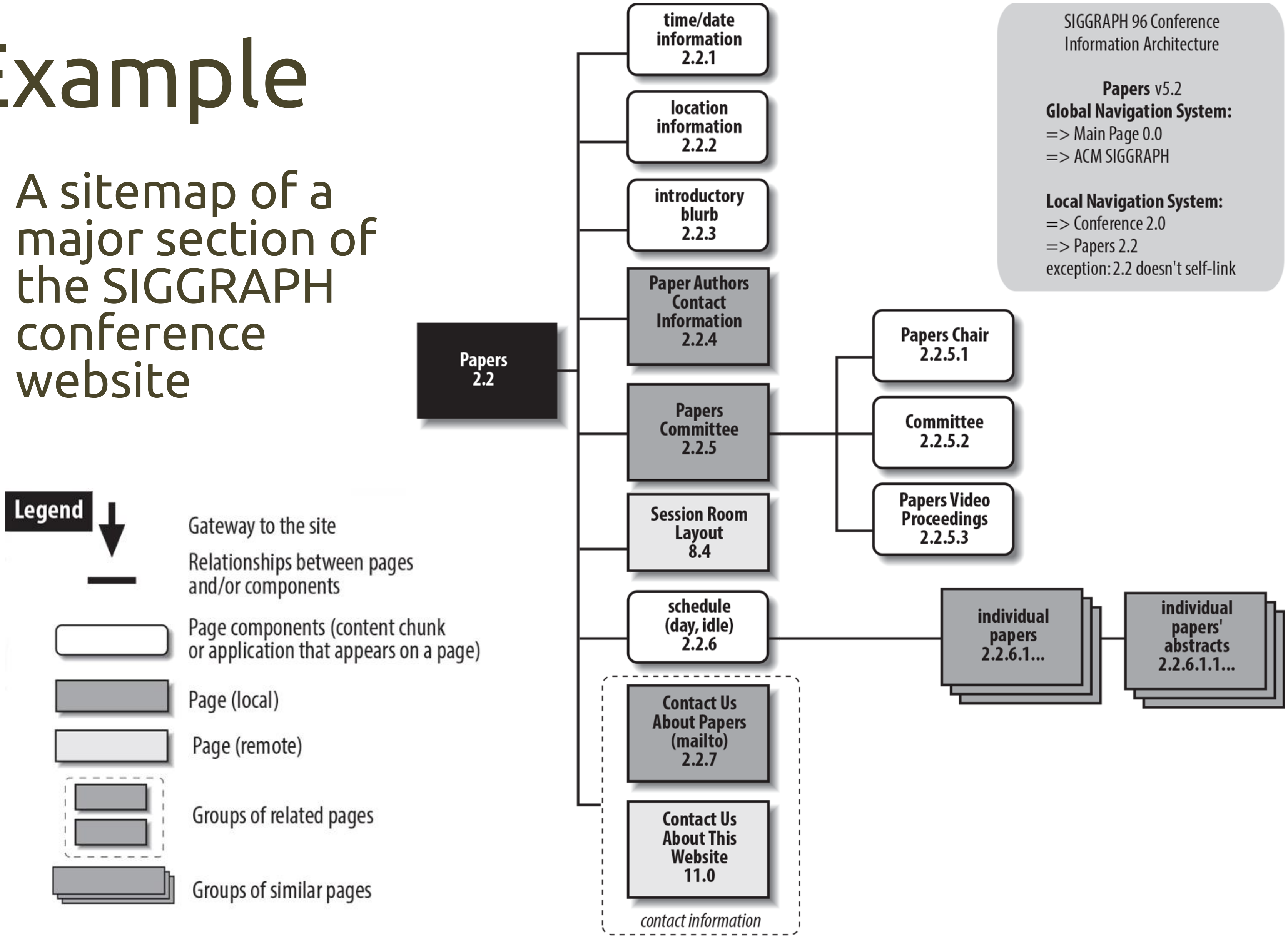


Detailed sitemaps

- As you move deeper into the implementation stage, your focus naturally shifts from external to internal
  - Rather than communicating high-level architectural concepts to the client, your job is now to communicate detailed organization, labeling, and navigation decisions to your colleagues of the development team

Example

- A sitemap of a major section of the SIGGRAPH conference website



Wireframes

- Sitemaps can help you determine where content should go and how it should be navigated within the context of a website, subsite, app, or collection of content
- Wireframes depict how an individual page or template should look from an architectural perspective
  - They connect the information architecture with its interaction design
- Wireframes are typically created for the product's most important pages or screens (such as main pages, major category pages, interfaces to search, …)
  - They are also used to describe templates that are consistently applied to many pages, such as content pages
  - They can be used for any page that is sufficiently confusing to merit further visualization during the design process

Example

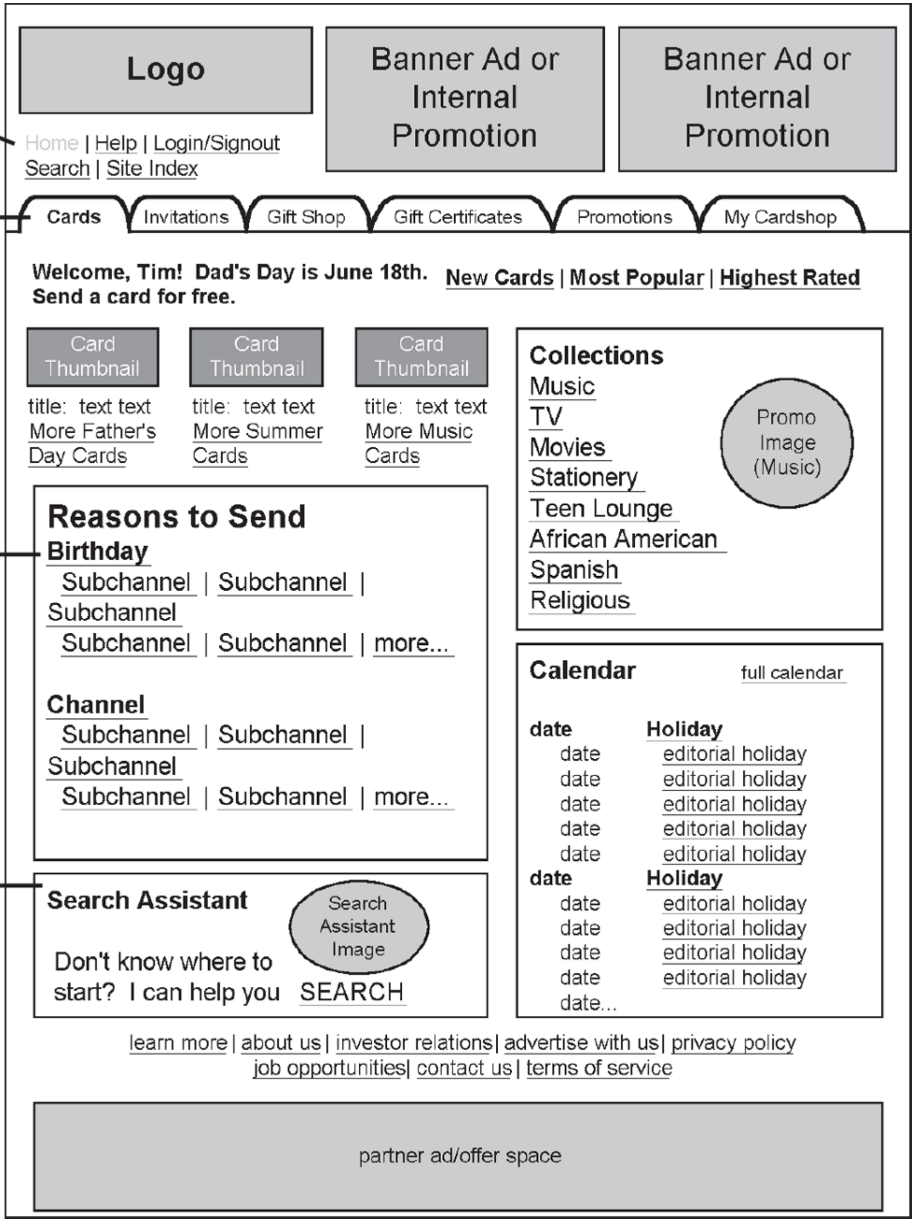
- The main page of a greeting card site

Header navigation for site-wide functions.

Tabs represent major categories of services.

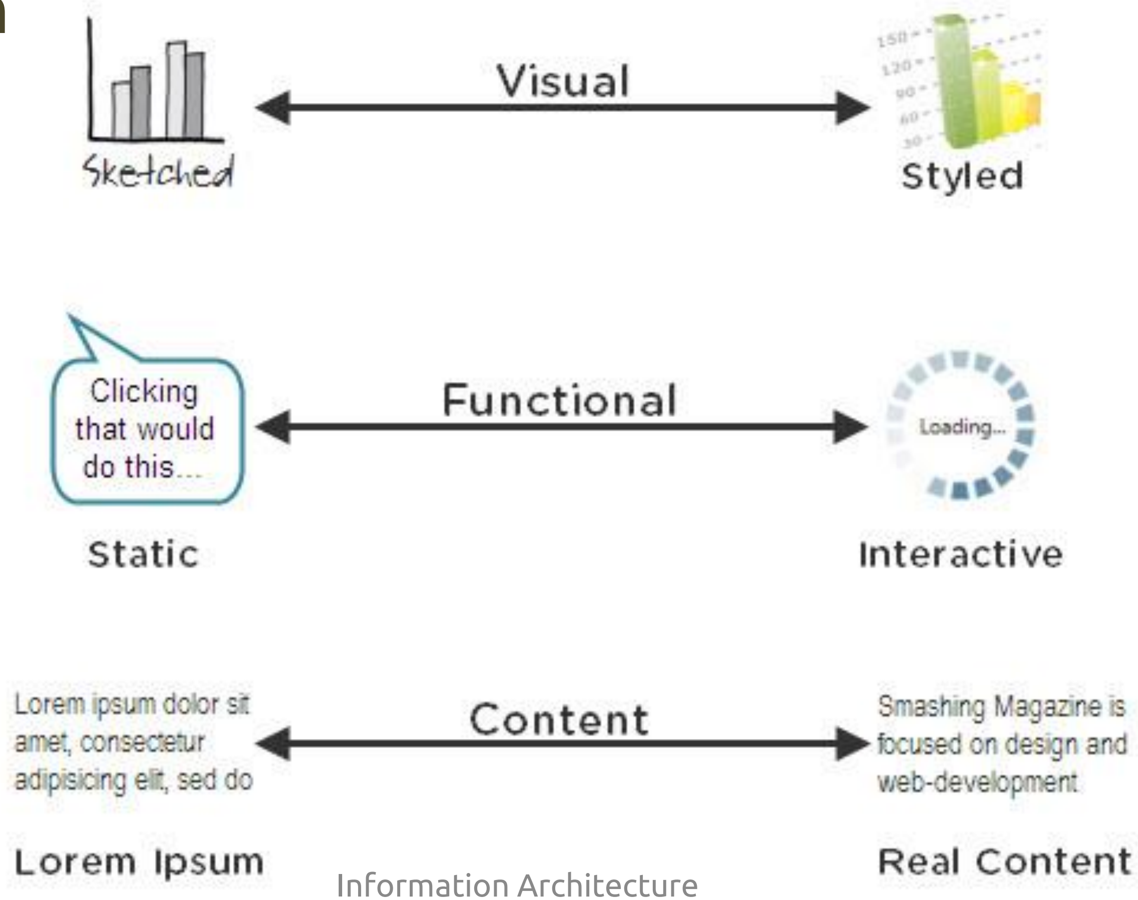
Primary card classification scheme. Expand level two channels as much as possible.

Promote searching using the wizard on home. Position to catch users not satisfied by channels.



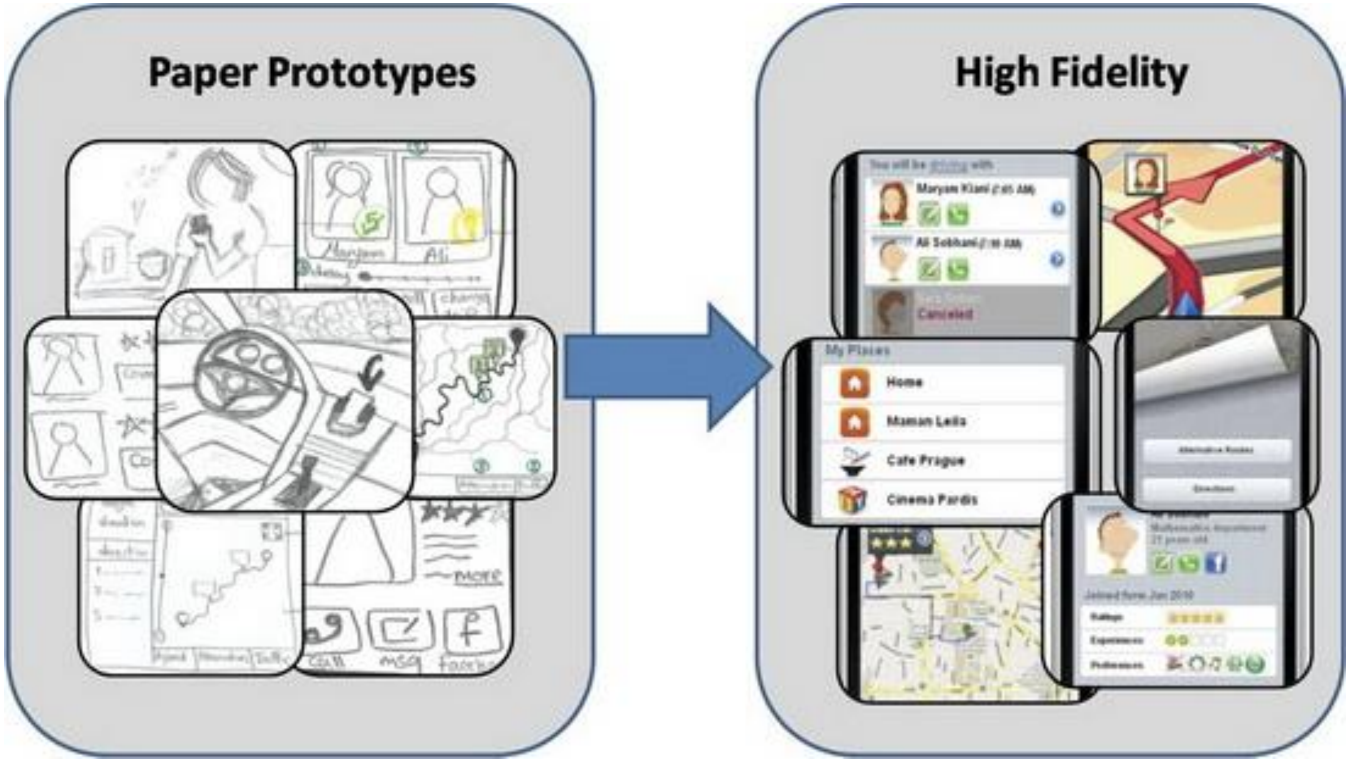
Interactive prototypes

- More than sketches or scenarios, these digital artefacts show how the product will look and function



Prototypes

- Low-fidelity vs high-fidelity prototypes



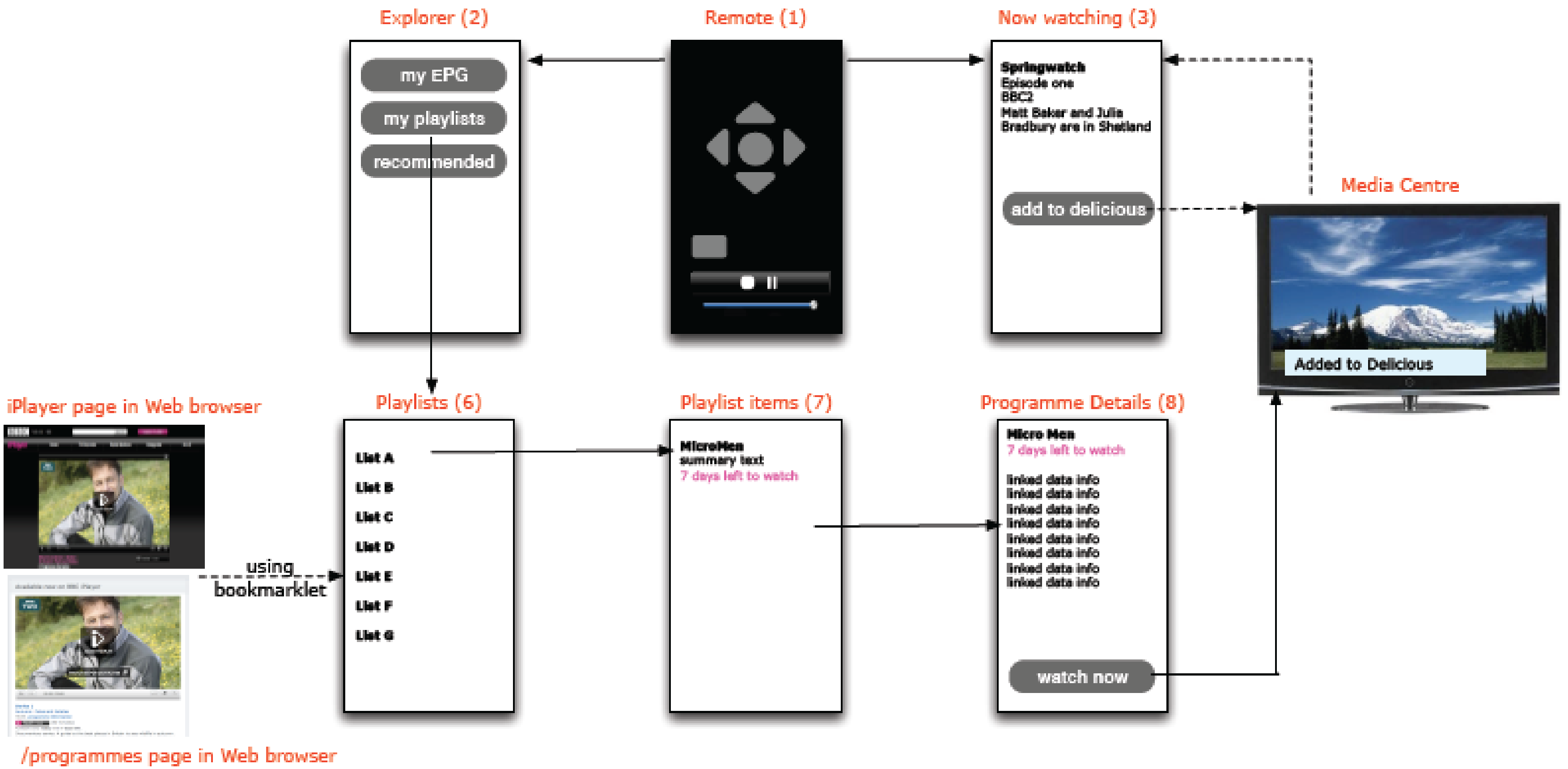
Low-fidelity prototypes

- Very coarse-grained
- Fuzzy layouts of general system requirements
- Paper-based and digital
  - Sketching
  - Screen mockups
  - Storyboards
- Used to gather feedback on the basic functionality or visual layout

Paper prototypes

- Sketches and screen mock-ups
  - Quick to build
  - Easy to run
- Storyboards
  - Sequence of screens focusing on a user action
  - Don't capture every detail, just systems' major functionality
  - Could be limited in scope, more rigidly linear
  - Users love paper prototypes
  - Opportunity to contribute to the new design

Wireframes



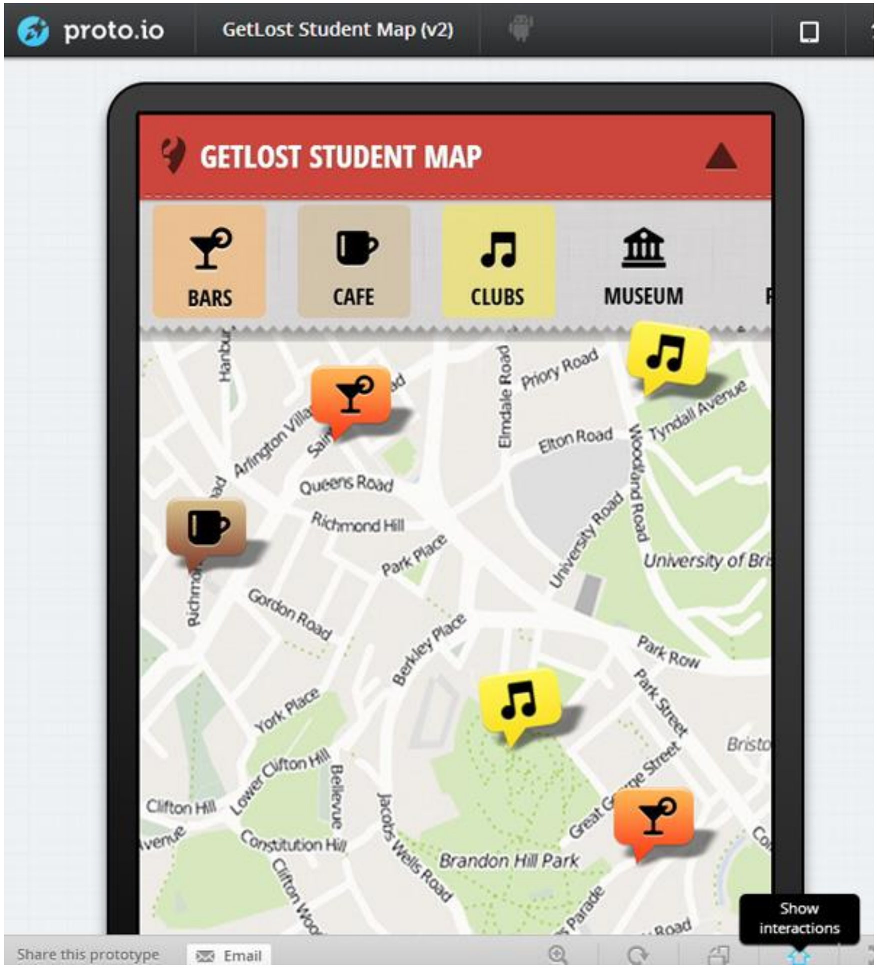
Explorer (2)
my EPG
my playlists
recommended
Remote (1)
Now watching (3)
Springwatch
Episode one
BBC2
Matt Baker and Julia Bradbury are in Shetland
add to delicious
Media Centre
Added to Delicious
iPlayer page in Web browser
using bookmarklet
/programmes page in Web browser
Playlists (6)
List A
List B
List C
List D
List E
List F
List G
Playlist items (7)
MicroMen
summary text
7 days left to watch
Programme Details (8)
Micro Men
7 days left to watch
linked data info
linked data info
linked data info
linked data info
linked data info
linked data info
linked data info
linked data info
watch now

Low-fidelity prototypes

Advantages	Disadvantages
They are cheap to produce. They can evaluate design ideas and design alternatives. They promote rapid, iterative development.	Their ability to check errors in design is limited. The specification is less detailed so it may be more difficult for programmers to code.
They are useful for facilitating communication between users and stakeholders and the UI designer.	A human facilitator is needed to simulate how the UI will work (e.g., by manipulating how different prototypes in response to users actions). Paper may seem less compelling.
They can show the look and feel and layout of screens.	They are useful for gathering requirements but are generally thrown away once the requirements have been established.

High-fidelity prototypes

- Fine-grained
- Highly elaborate and polished digital versions of the system
- Used to gather detailed information on the processes involved in traversing several parts of the system, or a subset of tasks



High-fidelity prototypes

Advantages	Disadvantages
They can show complete functionality.	They are more time consuming to create than low-fidelity prototypes.
They can show the look and feel, layout, and behavior of the final product.	They are not as effective as low-fidelity prototypes for requirements gathering, because they cannot easily be changed during testing.
They are fully interactive, and can be useful as a marketing tool (demo).	They can look so professional and finished that users are less willing to comment. This may mean that the prototype gets built irrespective of its merits and loses its throw-away benefits.

References

- Louis Rosenfeld, Peter Morville and Jorge Arango, “Information Architecture for the Web and Beyond, Fourth Edition”, O’Reilly

License