Module -3
User Experience Design

SYLLABUS
Frameworks for User Centric Computing, Computational models of users, Advancing the User Experience- Display Design, View (Window) Management, Animation, Web Page Design, Color. Timely user Experience-Models of System Response Time (SRT) Impacts, Frustrating Experiences, Information Search- Five Stage Search Framework, Data Visualization-Tasks in Data Visualization, Challenges

FRAMEWORKS FOR USER CENTRIC COMPUTING

User-centric computing is fundamentally concerned with **designing interactive systems with a deep understanding of and focus on the users, their tasks, and their environment**. Frameworks in Human-Computer Interaction (HCI) and user-centric design provide structured approaches, theoretical underpinnings, and practical guidance to achieve this goal.

Here are some key aspects of frameworks for user-centric computing:

1. **Structured Design Methodologies and Processes:**
    - Frameworks offer advice in various forms, such as steps, questions, concepts, challenges, principles, tactics, and dimensions, to constrain and scope the user experience.
    - **ISO 13407** provides guidance on **human-centered design processes** for interactive systems, focusing specifically on the management of the design process.
    - Principles laid down for a "useful and easy to use computer system" include an **early focus on users and tasks**, requiring direct study of user characteristics (cognitive, behavioral, anthropomorphic, attitudinal) and involvement of users in the design process.
    - Methods like **Contextual Design** involve understanding users by observing them in their work context, building work models, and keeping data about the customer and their work visible in a design room.
    - Other methodologies include the **Logical User-Centered Interactive Design Methodology (LUCID)** and **IBM's Ease of Use development methodology**, which outline stages or activities for user-centric design.
2. **Conceptual Frameworks and Theories of Cognition and Interaction:**
    - Applying **theories and conceptual frameworks to interaction design** is a significant approach. These often draw from theories of human behaviour.
    - **Cognitive frameworks** aim to explain and predict user behaviour.
    - Examples include:

        - **Mental models**: Representing a user's understanding of how a system works
        - **Information processing**: Viewing the user as processing information through perceptual, cognitive, and motor stages (e.g., the Model Human Processor)
        - **External cognition**: How people use external resources (like notes or other people) to support their thinking.
        - **GOMS (Goals, Operators, Methods, and Selection rules)**: A hierarchical model describing how users perform tasks
        - **Keystroke-Level Model (KLM)**: A simplification of GOMS for predicting task execution time.
        - Norman's seven stages of action: Describing the user's process from goal formation to outcome evaluation.
    - **Context-of-use theories** highlight the critical role of users' complex interactions with their physical and social environments, asserting that design cannot be separated from patterns of use.
    - Frameworks can help place different areas related to HCI, such as ergonomics, dialog design, and interface styles, in the context of the interaction. Ergonomics, for instance, addresses issues on the user side of the interface.
    - **Paradigms** for interaction (e.g., time-sharing, WIMP, direct manipulation, ubiquitous computing, CSCW, tangible bits) represent effective strategies and provide historical perspective on designing usable interactive systems.
    - The **CfA (conversation for action) framework** has been used as the basis for a conceptual model to facilitate communication in systems.
    - The **Xerox Star interface** is considered a classic conceptual model based on the familiar knowledge of an office environment, using icons to represent real-world objects.
    - **Direct Manipulation** is described as an interaction style framework where the system is an extension of the real world, allowing users to focus on data rather than the underlying system structure.
3. **Design Principles and Standards:**
    - Well-designed interfaces are guided by principles stressing consistency, predictability, and controllability, aiming to leave users feeling competent, satisfied, and responsible.
    - **ISO standards** provide formal guidance on ergonomic requirements, multimedia interfaces, human-centered design, and the ease of operation for everyday products, contributing to universal usability. **ISO/CD 20282** specifically focuses on meeting the needs of a wide range of users, including those with disabilities.
4. **Software Implementation Frameworks:**
    - **User Interface Management Systems (UIMS)** offer programming and design techniques with a **conceptual architecture** that separates application semantics from presentation.
    - **Application frameworks**, based on object-oriented programming, capture the typical structure of user interface programs, allowing applications to be built by completing the framework. Examples include MacApp, NextStep, Cocoa, and Microsoft Foundation Classes (MFC).

    - **Windowing systems** form a foundation, providing device independence and abstracting hardware specifics for programmers, although they offer only a crude level of abstraction for separating presentation and application dialog control.

In essence, frameworks for user-centric computing draw upon diverse knowledge from psychology, sociology, computer science, and design practice to provide structure and guidance throughout the design and implementation lifecycle, ensuring that the user remains the central focus. These frameworks help designers understand user capabilities and limitations, predict behaviour, structure interactive systems, and evaluate usability.

COMPUTATIONAL MODELS OF USERS

Computational models of users in user-centric computing provide structured ways to represent, understand, and predict user behaviour and interaction with systems. These models are often derived from psychological theories and aim to inform the design and evaluation of interactive systems.

Here are some key computational models and approaches discussed:

1. **Cognitive Models:** This is a major category of models aiming to represent users interacting with systems, focusing on aspects of their understanding, knowledge, intentions, or processing.

    - **The Model Human Processor (MHP):** This is a highly influential and classic cognitive architecture model in HCI. Derived from information processing theory, it offers a simplified view of human processing during interaction. It conceptualises cognition as a series of processing stages involving distinct perceptual, cognitive, and motor processors, each with its own memory. The MHP predicts the cognitive processes involved in user interaction and serves as a basis for calculating the time users might take to perform various tasks. It primarily focuses on internal cognitive processes.
    - **The GOMS Family of Models:** Developed from the Model Human Processor, GOMS (Goals, Operators, Methods, Selection rules) is a generic term for a family of models used to predict user performance.
        - **GOMS:** This model describes how a user performs a computer-based task in terms of specific goals they want to achieve, the operators (cognitive processes and physical actions) needed, the methods (learned procedures) for accomplishing goals, and selection rules used when multiple methods are available. It is particularly effective for modelling expert users performing routine tasks and can predict execution times and effective strategies. It represents interaction using a hierarchical structure of user goals and tasks.
        - **Keystroke-Level Model (KLM):** A simplification within the GOMS family, KLM focuses on predicting the execution time for relatively simple, short tasks (typically under 20 seconds). It's based on a detailed understanding of the human motor system and assigns time estimates to basic physical and mental operations (like pressing a key, pointing with a mouse, or thinking). It's applicable in situations where the user's actions are relatively automatic with little planning.

- **Linguistic Models:** These models view user interaction through the lens of language, aiming to understand user behaviour and analyse the cognitive difficulty of the interface. Examples include formalisms like BNF (Backus–Naur Form) and TAG (Task-Action Grammar), which are often used to specify user-system dialogs but can also model the user's understanding of the interaction grammar.
- **Physical and Device Models:** Building on the understanding of the human motor system, these models help predict performance related to physical actions. KLM is a key example here. Other models in this category include Fitts' law, which predicts the time taken to move to a target, and Buxton's three-state model for pointing devices.
- **Cognitive Architectures:** Models like the Model Human Processor, the problem space model, and ICS (Interactive Cognitive Subsystems) provide the foundational assumptions about human cognition upon which other cognitive models are built.
- **Cognitive Complexity Theory (CCT):** This theory can produce models of both user goals and the system grammar and reason about their interaction. These cognitive models often have a "computational flavour," reflecting the use of computational analogies in cognitive psychology. While they provide a powerful way to analyse and predict user performance, particularly for planned or routine tasks, they may rely on detailed system information or not fully account for complex, unplanned, or continuous interactions.

2. **User Modeling for Adaptive Interfaces:** Interactive systems often incorporate a model of the user, either implicitly in the design or explicitly within the program. Systems with a built-in, explicit user model can guide an adaptive interface. The system can keep track of user performance and adapt features like menu content, order, or feedback type to suit their needs, for instance, presenting advanced menu items as users become more proficient. This involves a computational representation of the user's state, proficiency, or preferences to enable real-time adjustments to the interface.

3. **Stages-of-Action Models:** Psychological models that describe the user's process when interacting with systems can inform computational representations.

   - **Norman's Seven Stages of Action:** This model describes a cycle including forming a goal, forming an intention, specifying an action, executing the action, perceiving the system state, interpreting the system state, and evaluating the outcome. While primarily a psychological model, its structured stages provide a framework for computationally modelling the user's interaction process.
   - **Simple Action Cycle Models:** Models depicting the sequence of user initiation, computer response, and user thinking/planning time. These focus on the timing and flow of interaction, which can be computationally analysed to understand and predict performance delays.

4. **Mental Models:** These refer to the representations users construct in their minds about themselves, others, objects, and the environment, used to understand situations and decide actions. While often discussed conceptually, frameworks like cognitive complexity theory computationally reason about the relationship between user goals (part of a mental model) and system structure.

More recent model types mentioned include user models that predict users' information needs and models characterising core components of the user experience. These models draw on diverse sources, often abstracting theories from psychology and other contributing disciplines to provide a simplified, structured view of user behaviour or aspects relevant to HCI design.

ADVANCING THE USER EXPERIENCE

Advancing the user experience in human-computer interaction is closely tied to thoughtful design across various interface elements. Key areas where design efforts contribute significantly to the user's perception, performance, and satisfaction are **Display Design, View or Window Management, Animation, Web Page Design, and Colour**.

1. Display Design

Display design is fundamental to how users perceive and interact with a system. The layout of information on a display offers opportunities for design improvements. Cluttered displays can overwhelm users, but well-organized, information-abundant layouts can reduce search time and increase subjective satisfaction. Design of displays involves considerations from psychology and graphical design, aiming to present information clearly and serve as the locus for interaction. Basic principles at the screen level reflect those in other areas of interaction design.

Key aspects for effective display design include:

- **Using high-resolution monitors** and maintaining them for maximum display quality.
- Designing layouts with **physical grouping, ordering of items, decoration** (like fonts, lines), **colour, alignment, and the use of white space** to emphasize the logical structure of interaction.
- Ensuring **information architecture can be understood** even when display elements change due to user control of font size, window size, and brightness.
- Ensuring that the **physical structure of the screen emphasizes the logical structure** of the user interaction.
- Presenting **information appropriate to the user's purpose**.
- Allowing **user control of data display**, such as easily changing the order of columns and sorting of rows. This supports getting information in the form most convenient for the task.
- Ensuring controls have **appropriate affordances** – visual and tactile attributes suggesting their use.
- Providing **meaningful contrast between screen elements**.
- Creating **spatial groupings** of elements.
- **Aligning screen elements and groups**.
- Providing **three-dimensional representation** where appropriate.
- Presenting the **proper amount of information on each screen** – too little is inefficient, too much is confusing. Ideally, all information necessary for an action or decision is on one screen.
- Dividing information into **logical, meaningful, and sensible units**.
- Providing an **ordering of information** that is prioritized according to user expectations/needs, logical, sequential, rhythmic (guiding the eye), encourages natural movement, and minimizes pointer/eye movement distances.
- Using layout to **reduce eye and hand movement**.

- Assisting users in **navigating through a screen** by aligning elements, grouping elements, and using line borders.
- Providing a visually pleasing composition through **adequate use of white space**, balance, groupings, and alignment.
- Avoiding visual clutter by maintaining **low screen density** levels and element distinctiveness.
- Providing **visual emphasis** to the most important screen elements.
- Considering **aesthetics**, though it may conflict with utility, as good graphical design is a skill of its own. Bad use of colour and 3D effects are bad for both aesthetics and usability. Research suggests that aesthetics influences perceived ease of use and can create a positive attitude, making users more tolerant of problems and aiding creative thinking/problem solving.

Display design guidelines were an early topic in human-computer interaction research due to their importance in critical applications. As technology evolved, new empirically validated guidelines became necessary. There is renewed interest in display design with the advent of small, wall, and mall-sized displays. Innovative information visualizations with user interfaces that support dynamic control are a rapidly emerging theme. These tap into the remarkable human perceptual ability for visual information, allowing relationships to be shown by proximity, containment, lines, or colour coding, and highlighting to draw attention.

2. View or Window Management

Effective multiple-window display strategies can be developed. **If window-housekeeping actions can be reduced, users can complete tasks more rapidly and probably with fewer mistakes**, which significantly advances the user experience by increasing efficiency and reducing frustration. The visual nature of window use has led designers to apply **direct-manipulation strategies** to window actions. Users can point at icons on the window border and click/drag to stretch, move, and scroll windows.

While window-management strategies have become standardized, there are **opportunities for innovation** with large and multiple displays, novel applications requiring coordination, and work-management strategies. Coordinated windows represent a potential next generation of window managers. These windows can appear, change contents, resize automatically, and close as a direct result of user actions in the task domain, streamlining workflows like processing medical claims. Such sequences can be established by designers or users with end-user programming tools.

The dynamics of windows have a strong effect on user perceptions, meaning that the animations for transitions must be designed carefully. Window design methods, when executed well, can enhance the user experience.

3. Animation

Small animations are now common in many applications to create an attractive look and feel. Perceiving and interpreting motion is a fundamental element of human perception, and our eyes are attracted to moving objects. Done well, animation is compelling. However, if done poorly, **animation can distract, irritate, and waste time**, and users almost universally disapprove of techniques like blinking advertisements.

Animation is appreciated primarily when it provides meaningful information. Possible roles for animation include:

- Keeping the user **oriented during transition** (e.g., zooming, panning on maps, opening/closing windows).
- Indicating an **affordance, inviting interaction** (e.g., a flashing down arrow encouraging scrolling).
- **Entertaining**.
- Indicating **background activity** (e.g., a progress bar).
- **Storytelling**.
- **Alerting**.
- Providing a **virtual tour** (e.g., for architectural designs).
- Explaining a **process**.
- Conveying **uncertainty and randomness**.

Smooth animations are now expected to keep users oriented during screen layout reorganizations. Individual screen elements can use animation to guide users. Showing changes over time is a natural application and particularly effective when coupled with voice narration. The dynamics of windows and the animations for transitions (zooming boxes, repainting, blinking outlines, highlighting during dragging) must be designed carefully as they have a strong effect on user perceptions. Animated icons can also be used as a way of conveying expressivity, such as a recycle bin expanding.

4. Web Page Design

The steadily growing interest in the World Wide Web stems from a desire to improve the user experience. Web page design has evolved from primarily focusing on navigation and presenting information (content, not data) to also encompassing the design of web applications. Web page design is largely about properly **balancing the structure and relationships of menus, content, and other linked documents or graphics** to build a hierarchy that feels natural, is well structured, easy to use, and truthful. The goal is to enable users to navigate complex information structures easily.

Web page designs are improving as standards and tools emerge. Key aspects for advancing the web user experience include:

- Achieving designs that are **not only usable but also aesthetically pleasing**. Getting the graphical design right is critical.
- Using **graphical elements** (background images, colour, bold text, icons) to make a website distinctive, striking, pleasurable, and readily recognizable. However, there is a danger of getting carried away with appearance at the expense of findability and navigation.
- Using **HTML or CSS for smooth animations and sleek graphic design**, turning basic menus into custom widgets that define the look and feel.
- Designing links, menus, choices, and commands using **familiar terminology or recognizable visual elements**.
- Organizing elements in a **meaningful structure and sequence**.
- Enabling users to navigate easily with clicks, taps, or smooth scrolling.
- Considering **user preferences** as crucial, especially if they demonstrate that visitors stay longer and buy more at visually compelling sites.
- Addressing issues of **download times**, which can be a downside of graphics.

- Supporting user interaction approaches like **browsing** (non-specific surfing, wandering at whim). A person's web experience is shaped by how well the design reflects their expectations, capabilities, limitations, visual appearance, content, and functionality.
- Considering criteria such as the **visual design style** (reflecting purpose, compatible with user tasks/needs), **content legibility/readability**, **backgrounds/color** (compatible, not distracting), and **graphics/icons** (meaningful, efficient, not distracting).
- Minimizing **functional barriers** (slow pages, excessive ads, illegible fonts, confusing organization) and **aesthetic barriers** (noisy screens, overuse of color).
- Providing **points of prospect** where users quickly get oriented and can survey options (scannable pages, orientation cues, clear navigation).
- Ensuring that all elements have **meaning to users** and serve a purpose.
- Supporting the ability for **user-generated content** to be easily and rapidly inserted into websites.
- Promoting **universal usability** through web design guidelines.
- Designing the web for how **users behave**, with designers taking responsibility for content and usability. This involves tasks like writing headlines that make sense and designing navigation.
- Exploring new metaphors beyond the "page metaphor" to help users manage information overload and prioritize time on the web.

Research areas include increasing designer knowledge of web tools and methods to advance user experience, facilitate user-generated content, and promote universal usability. Trendy designs like parallax scrolling need research to determine if they are easy/fun or confusing.

5. Colour

Color displays are attractive to users and can often improve task performance. However, the danger of misuse is high. Improper use of color can mislead, distract, and slow users.

Color can contribute to the user experience and task performance in several ways:

- **Soothe or strike the eye**.
- **Add accents** to an uninteresting display.
- Facilitate **subtle discriminations** in complex displays.
- Emphasize the **logical organization** of information.
- Draw **attention to warnings** or important information.
- Evoke **strong emotional reactions** of joy, excitement, fear, or anger.
- Assist in **formatting** a screen by relating/breaking groupings and associating information.
- Serve as a **visual code** to identify screen components, logical structure, sources, or status.
- Realistically **portray natural objects**.
- Increase **screen appeal**.

Principles from graphic artists have been adapted for user interfaces. Designers are learning how to create effective computer displays and avoid pitfalls. In monochromatic design, colour can be judiciously added where it aids the operator. For getting the user's attention, colour is listed as a technique, suggesting using up to four standard colours with additional ones reserved for occasional use.

Guidelines for using colour include:

- Using color as an **enhancing design characteristic**.
- Selecting **color combinations that can be discriminated**.
- Ensuring high **lightness contrast** between foreground and background color.
- Increasing lightness contrast between colours in the color spectrum (blues and reds).
- Avoiding combining dark colors from the middle of the spectrum with light colors from either end of the spectrum.

Colour usage guidelines exist on the World Wide Web, and browsing lively and colorful web sites can be an informative and enjoyable experience. Color can improve some displays and lead to more rapid task performance with higher satisfaction. However, overuse of colour can be an aesthetic barrier.

In summary, advancing the user experience involves dedicated design efforts across the interface, leveraging techniques in display layout, window management, animation, web page structure and aesthetics, and the careful use of colour to enhance clarity, efficiency, engagement, and overall user satisfaction.

TIMELY USER EXPERIENCE

The concept of the "timely user experience," or Quality of Service, is a significant concern for users and providers of interactive systems. It stems from the basic human value that **time is precious**. When users experience externally imposed delays, they can become frustrated, annoyed, and even angry. Inadequate performance, such as lengthy or unexpected system response times and slow display rates, leads to these reactions, potentially causing frequent errors and low satisfaction. Apps may be abandoned quickly if performance is inadequate, and customers might switch to competitors with better user experiences.

Models of System Response Time (SRT) Impacts

System Response Time (SRT), often simply called "response time," is defined as the number of seconds from when a user initiates an action (like pressing Enter or clicking a mouse button) until the computer begins presenting results or feedback. After the system completes its response, the user begins formulating the next action, which takes a duration known as "user think time".

A simple model depicts a sequence where the user initiates an action, waits for the computer's response, watches the results appear, processes them, thinks, and then initiates the next action. However, a more realistic model shows users planning while interpreting results, while inputting actions, and even while the computer is processing or retrieving information. Users tend to use any available time to plan ahead, making precise measurements of user think time difficult. The computer's response is generally more precisely defined, though some interfaces might provide immediate minor feedback before the main results appear after a delay.

Designers and network managers considering SRT must account for a complex interplay of factors: technical feasibility, costs, task complexity, user expectations, speed of task performance, error rates, and error-handling procedures. These decisions are further complicated by individual user differences like personality, fatigue, familiarity, experience, and motivation.

SRT significantly affects user behaviour, learning, and performance.

- **Planning and Memory:** If steps in a user's plan produce unexpected results or if delays are long, users may forget parts of the plan or need to review it repeatedly. Long response times can lead to wasted effort and more errors due to repeated plan reviews.
- **Pace and Errors:** Users often pick up the pace of the interface as response times shorten. While shorter SRTs can lead to higher productivity, especially for simple, frequent tasks, they can also generate a faster pace where solution plans are prepared hastily, potentially increasing errors, particularly for complex tasks. This reflects a speed/accuracy tradeoff. The optimal pace depends on the task and the cost and ease of error recovery.
- **Anxiety:** As response times grow longer, the penalty for errors increases, which can raise user anxiety and further slow performance and increase errors. However, anxiety about an erroneous command might also emerge if the response time is unusually short.
- **Task Type:** For simple, repetitive tasks, users generally prefer rapid performance and are annoyed by delays over a few tenths of a second. For complex problems, users are more adaptable to longer delays and can use the time to plan ahead.
- **User Experience:** Novices may perform better with slightly slower response times and may prefer working at a slower pace than experienced users. More experienced users tend to work faster and may prefer shorter response times. They will adapt their working styles and strategies as SRTs change.

Variability in Response Time: Users expect predictable behaviour. Extreme or unexpected variation in response time can be unsettling. Modest variations (around plus or minus 50% of the mean) seem tolerable and have little effect on performance. However, delays that are unusually long (at least twice the anticipated time) can cause frustration, and anxiety may arise from unusually short response times (less than one-quarter of the anticipated time). Some studies suggest that reducing variability in response times can improve user performance. Designers should strive to avoid extremely slow responses or provide progress indicators if delays are necessary. Slowing down unexpectedly fast responses is a controversial suggestion to avoid surprising users, affecting only a small fraction of interactions. Providing progress reports or indicators for slow responses, such as a large clock ticking backwards, page numbers, or percent complete, confirms the system is working.

General Response Time Guidelines based on research:

- Typing, cursor motion, mouse selection: 50-150 milliseconds
- Simple, frequent tasks: 1 second
- Common tasks: 2-4 seconds
- Complex tasks: 8-12 seconds
- Users should be advised of long delays.
- Rapid start-ups are important.
- Modest variability is acceptable, but unexpected delays are disruptive.
- Offering users a choice in the pace of interaction is appealing.

Frustrating Experiences

While technology promoters argue that the quality of user experiences has improved with increased speeds and capacities, critics contend that frustration from interface complexity, network disruptions, and malicious interference has grown. Recent research aims to document and understand the sources of user frustration with contemporary interfaces.

Common sources of user frustration include:

- **Long delays and slow response times:** This is a significant cause of frustration. Users want to work quickly, and waiting too long can test memory limitations and impair productivity. Uncertainty about how long a delay will be can be a greater source of annoyance than the delay itself.
- **Application crashes that destroy data:** Crashes are cited as a source of frustration.
- **Software bugs that produce incorrect results:** Bugs also contribute to frustration.
- **Poor designs that lead to user confusion:** Confusing error messages and overly complex interfaces are significant frustration provokers. Messages that are too general make it difficult for novices to understand what went wrong. Requiring users to retype entire records after a minor error is also frustrating.
- **Networked environment issues:** Unreliable service providers, dropped lines, and network outages are frustrations. Spam (unsolicited e-mail) and malicious viruses are also serious threats and frustrations.
- **Unexpected behaviour:** Adaptive systems that make surprising changes can puzzle and frustrate users because they cannot predict the next change, interpret what happened, or restore the previous state. Unsettling variations in response time can also cause apprehension.
- **Interruptions:** Whether from the current task or unrelated ones, interruptions are troubling, even though users might compensate by working faster, which comes at the price of more stress and higher frustration.
- **Excessive housekeeping tasks:** This is mentioned as a classic frustration provoker.
- **Lack of human contact in automated systems:** Users dealing with complex menu systems where options are unclear often long for a human operator.

User surveys and studies have documented high levels of frustration. One British study found nearly half of office workers felt frustrated or stressed by the time it takes to solve problems. An American survey estimated 5.1 hours per week of wasted time due to computer issues. Another study found 46-53% of users' time was seen as wasted, with frequent complaints including dropped network connections, application crashes, long system response times, and confusing error messages. Studies of student and workplace users show high frustration levels and significant wasted time.

Reducing user frustration involves addressing these issues through various means, such as improving server capacity and network speed, better user training and online help, redesigning instructions and error messages, and protecting against spam and viruses.

In essence, the timely user experience is fundamentally about managing system response times effectively to match user expectations and cognitive processes, and by doing so, minimizing the wide array of frustrating experiences that can stem from poor design, system issues, and unpredictable behaviour.

INFORMATION SEARCH

Information Search is a fundamental activity in Human-Computer Interaction, particularly in the context of accessing vast digital information spaces like the World Wide Web, digital libraries, and databases. It's a core component of user experience, aimed at helping users find, filter, and make sense of available information.

Terminology and Related Concepts

The domain of finding information has a "colorful" swirl of terminology. Older terms include **information retrieval**, often applied to bibliographic and textual document systems, and **database management**, for structured relational database systems. These are being supplemented or replaced by newer notions such as **information gathering**, **information seeking**, **information filtering**, **information visualization**, **collaborative filtering**, **sensemaking**, and **visual analytics**. Computer scientists focus on large volumes of data, using terms like **data mining** or **deep learning**. While distinctions are subtle, common goals range from finding a narrow set of items satisfying a specific need (a **known-item search**) to making sense of information or discovering unexpected patterns (exploratory search). Users often alternate between various strategies, a behaviour sometimes called **berry picking**. Information seeking also differs from **re-finding** something previously seen, where users might prefer navigation over search, especially if they don't remember specific keywords.

The Importance of Information Content

On the Web, the most sought-after commodity is **information content**. High-tech features, fancy graphics, and colours don't compensate for insufficient or poor content. Innovations are judged by how well they support the information presented.

Types of Search Tasks

User tasks can be decomposed into **browsing** or **searching**. Task complexity can vary:

- **Specific fact finding** or **known-item search**: Where there's a single, readily identifiable outcome. Examples include finding Hilary Clinton's email address or the highest-resolution satellite image of a specific location and time. Also, finding houses for sale with specific criteria or how to change a flat tyre.
- **Extended fact finding**: Tasks with uncertain but replicable outcomes. Examples: finding other books by an author or comparing economic indices between regions.
- **Unstructured tasks** (also called **exploratory searches**): These include exploration of the availability of information on a topic, open-ended browsing of known collections, and complex analysis of problems. Examples: exploring genealogical information at the National Archives or researching new work on voice recognition. Or analysing the role of women in historical photo collections or looking for new medical treatments.

When users have clear goals and know exactly what they want, they try to match this with available information. If unsure, they may browse, letting information guide their attention.

The Search Process: A Five-Stage Framework

Information seeking is often an iterative process. A widely discussed framework describes the search process in five stages:

1. **Formulation**: Expressing the search. This involves identifying the source(s) of information, specifying fields to limit the search, and defining phrases or variants. Users may prefer to limit searches to specific collections or fields (like title or abstract). This limitation, called **scoping**, can improve results but also lead to failures if the constraint is forgotten. Clearly displaying the source is important. Supplemental finding aids like tables of contents, indexes,

or subject classifications can help clarify needs. Understanding tasks and potential requests helps provide hot-topic lists and useful classification schemes. Conversion of user needs (stated in task-domain terminology) to interface actions is considered a large cognitive step.

2. **Initiation of action**: Launching the search. Actions can be explicit (initiated by buttons like "Search") or implicit (initiated by parameter changes that update results immediately). Auto-complete can guide users to successful or past queries.
    a. The initiation of action is the second stage in the search process, which can be either explicit or implicit.
    b. Explicit initiation often involves clicking a search button or pressing the Enter key on a keyboard.
    c. The magnifying glass icon has become a standard symbol for search, especially in interfaces with limited space.
    d. In spoken interactions, initiating a search may simply involve pausing or indicating verbally.
    e. Implicit initiation is an alternative approach, where any change to the search criteria immediately triggers a new set of search results.
3. **Review of results**: Reading messages and outcomes. Search terms and constraints should remain visible. An overview of results (e.g., total number) is helpful. Results can be categorized using metadata. Descriptive previews or snippets of each item should be provided. Search terms should be highlighted in the results. Allowing examination of selected items and providing visualizations (maps, timelines) where appropriate are useful. Users should be able to adjust the result set size and displayed fields, and change the sequencing (alphabetical, chronological, relevance ranked, etc.). Presenting 10 to 50 returns per page can optimise performance and preference. Users typically peruse only the first page of search results. Polite and useful messages should be presented, especially if no items are found or parameters are incorrect. Providing information or simple actions directly within the results page, when possible, is beneficial.
    a. The third stage involves the review of search results, which can be presented in textual lists, geographical maps, timelines, or other specialized visual formats.
    b. Clear indication of search failure is important to prevent users from abandoning their search.
    c. Typically, only about 20 results are initially displayed in a list, but larger sets are preferable for users with high bandwidth and large displays.
    d. Previews, such as carefully selected text samples or snippets, human-generated abstracts, photos, or automatically generated summaries, help users select a subset of results and refine their queries.
    e. Users should have control over how results are sequenced, such as alphabetical, chronological, relevance-ranked, or by popularity, to improve effectiveness.
    f. Highlighting search terms in the preview helps users evaluate the value of results, while previous visits and URLs provide additional context for assessing trustworthiness.
    g. Additional previews and overviews can facilitate browsing of results, such as graphical overviews indicating scope, size, or structure, and previews of samples from collections.

h. When dealing with a large number of results, providing an overview of the number of items in available categories helps users navigate and filter the results effectively.
i. Access to the full document is usually necessary for users to identify items of interest, with highlighting of search terms for easier navigation within the document.
j. Automatic scrolling to the first occurrence of a keyword and markers along the scrollbar indicating other occurrences can aid in navigating large documents.
k. Care should be taken to ensure that the search box does not obscure the location of terms found within a document, as this can hinder the review process.

4. **Refinement**: Formulating the next step. Search interfaces should provide meaningful messages and support **progressive refinement**. This involves changing search parameters (like phrases, time range, location) to narrow or expand the results. Suggestions for error correction (e.g., misspellings) can be proposed. If multiple phrases were used, items containing all phrases should be shown first, followed by subsets. If no documents are found, that failure should be indicated. Making parameter changes convenient is crucial. Providing related searches can help. Keeping search terms active and easy to clear facilitates progressive refinement. Query previews can also help eliminate undesired items and improve satisfaction.
5. **Use**: Compiling or disseminating insight. Users should be able to save and annotate queries, settings, and results. Results can be sent via email or used as input to other programs. Collecting explicit feedback (ratings, reviews) can be explored. There are three kinds of search histories that can be saved: search entry (terms), search results (lists), and user-saved items. Saving search entries (terms) is useful for auto-complete and switching between tasks. Saving results lists automatically is helpful for reviewing earlier results. User-saved items are for long-term retention.

Search Interfaces and Strategies

Providing powerful search capabilities without overwhelming novice users is a challenge. This is usually addressed by offering a **simple search interface** with a single input field and a button, along with a link to an **advanced search interface**. Advanced interfaces provide controls for composing Boolean queries, restricting scope, and applying filters using checkboxes, text fields, and menus. Expert users may need a wide range of tools and options to compose, save, replay, and revise elaborate query plans.

Different methods exist for specifying the search:

- **Keyword Queries**: Simple key-phrase queries yielding web-page results are effective. Users often use specific terms. An internal glossary and thesaurus can be useful for large sites.
- **Phrases**: Using phrases can be more accurate than single words, especially for names. Users should have the option to expand a search by breaking phrases into separate words.
- **Variants and Partial Matches**: Interfaces can support variations (case insensitivity, phonetic variations, abbreviations, multiple formats) and partial matches.
- **Boolean Operations and Proximity Restrictions**: These combining strategies can be specified by users or service providers. Structured controls (checkboxes, lists) or text boxes can be used for input.

- **Templates**: Pre-defined keywords that guide users in selecting search terms. Organising them hierarchically can restrict initial search sets. Users using templates have been found to find significantly more target sites.
- **"Natural" Language Queries**: Appealing to users, especially with spoken interaction. While computer understanding is limited, large datasets allow finding answers through query expansion, tracking user interactions, and statistical methods. The semblance of natural language is often achieved because other users have asked the same question and human answers are retrieved. However, habitability remains a problem.
- **Command Languages**: Although often hidden under front ends, languages like SQL remain widespread for searching structured databases. They allow expert users to write precise queries. Command languages can be used in combination with menus, allowing users to transition smoothly from novice to expert.
- **Dynamic Queries**: Visual search interfaces enhanced with implicit initiation and immediate feedback become powerful dynamic query interfaces.
- **Faceted Search**: Integrates category browsing with keyword searching. It uses hierarchical faceted metadata presented as simultaneous menus and dynamically updated counts as previews. This allows users to navigate along multiple conceptual dimensions, progressively narrowing or expanding the scope. It organises results in a recognizable structure and gives control/flexibility over metadata use. Dynamic counts help avoid empty result sets.
- **Visual Searches**: Using specialized visual representations (calendars, maps, treemaps) simplifies query fields and provides context, helping users refine their needs. They can provide information about data availability and a feeling of thoroughness. Adding abstract data previews and overviews transforms them into potent information visualization and exploration tools, allowing visual exploration before a search is specified. The widely cited principle for visual information seeking is "Overview first, zoom and filter, then details on demand".

Browsing vs. Searching

Browsing and searching are distinct but often interleaved activities.

- **Browsing**: Non-specific surfing where users move through information spaces at their own pace, following links, scanning headings, and using page cues. It's analogous to shopping and involves an exercise of recognition. Browsing tends to be more effective than searching for finding content and related content, requiring fewer clicks and allowing users to stay longer, dig deeper, and view more pages. It works better for items that are not specific or unique. Users often prefer to browse as it requires less effort and provides more context. Websites should facilitate scanning, provide multiple layers of structure, easy navigation, and freely use links to support browsing.
- **Searching**: A more structured mechanism where keywords are entered, and results are presented. It's partially an exercise in recall. Searching is generally preferred for finding something specific or unique.

Research suggests that well-organized and clearly presented information hierarchies can be as effective and satisfying as search engines. On sites with clear labels and many navigation options, people tend to browse rather than search. Users often turn to a search facility only when browsing fails. Therefore, a well-designed site with good organization, clear navigation links, and a site map is the most effective searching tool, even more so than the search facility itself. Nonetheless, a search facility should always be included on key pages. Searching and browsing should be closely integrated.

Problems and Frustration with Search Facilities

Despite the increasing sophistication of search, many users experience frustration. A survey found that 71 percent of users were frustrated by Web searches, and 46 percent found them nerve-racking. Issues stem from three areas:

- **Not understanding the user**: There's no single ideal search interface due to user variation. The designer may not anticipate the user's expertise level, the nature of every possible query, or the type of information sought. Basic human traits are often not considered, such as users not reading instructions, preferring to "try it," or forgetting the location of the search engine.
- **Difficulties in formulating the search**: Users may not know how to use or format a search query. This can be due to the diversity of user needs or inconsistencies across interfaces. Users are often expected to think like the programming behind the search engine. Choosing the wrong parameters can lead to no results.
- **Presenting search results**: The way results are presented can add to frustration.

Effective search services are judged on how easily users can find what's needed and how well confusion and frustration are reduced. The purpose should be to return *information* (answers) rather than just data.

Guidelines for Search Facility Design

Key guidelines for designing search facilities include:

- **Know Your Search User**: Identify expertise, common terms, anticipate needs (query nature, kind of info, data type, result amount), plan for switching purposes and flexibility. Match sophistication to users, application, and workflow.
- **Express the Search**: Allow simple searches, search the entire site (or clearly indicate otherwise), structure to site info/user needs, integrate search/browse.
- **Placement**: Make the search facility prominent on the homepage and include it on every page.
- **Methods**: Permit users to specify search extent (section, site, sources, global). Offer different ways to search: templates, keywords, variants, phrases, partial matches, synonyms. Provide internal glossary/thesaurus for large sites. Include a spell checker.
- **Controls**: Provide a text box (at least 20 characters), structured controls (checkboxes, lists), and a labeled command button ("Search") next to the text box.
- **Interfaces**: Provide separate simple and advanced search interfaces, with a link to the advanced one.
- **Guidance**: Provide clear instructions, online help, or a search wizard.
- **Progressive Search Refinement**: Allow users to control the size of the result set. Support a rapid rough search showing item count or topical list, followed by a refinement phase.
- **Present Meaningful Results**: Goal is to provide exactly the info/answer in an easy format. Present a summary of criteria with results. Provide meaningful messages. Present concise (10-50 returns preferred), relevance-ordered, clear, scannable textual listings. Allow modifying sequencing and clustering results. Use graphical listings (maps) if clearer. Avoid

overwhelming detail like URLs. Include page upload dates only if relevant. If no items found or parameters are incorrect, provide polite, useful messages.
- **Remember the Search**: Save search entries, results automatically, or allow users to save results.
- **Destination Pages**: When linking to a page from search results, describe how the page relates to the query (summary) and highlight keywords.
- **Locatability**: Use text-based content, repeat keywords, and craft meaningful page titles.

Specialized Search Areas

- **Textual Documents and Databases**: Strategies involve parsing text grammatically, using synonyms, handling variations and misspellings, and separating queries into components. SQL is a standard for relational databases.
- **Multimedia Documents**: Includes search for images (QBIC - querying by image content, searching features or colours), sound, and video. Requires integrated tools for annotation, indexing, algorithms, and media-specific browsing. Video search can involve browsing collections along dimensions. Speech and audio search is emerging.
- **Geographic Information**: Increasingly used, with queries based on location and direction. User interfaces often include map displays for context.
- **Social Aspects**: "Social search" is an umbrella term. **Collaborative filtering** and recommenders can sometimes eliminate the need for a search box. Current **personalization** builds user profiles automatically from history and compares them for suggestions. This can be a form of implicit search. Better understanding of social and personalized search is needed.

Multimedia Document Search and Specialized Search

1. Image Search:
    - Relies on text searches in descriptive documents, keywords, tags, and metadata.
    - Challenges include finding specific images without proper tagging or captioning.
    - Automatic tagging and collaborative tagging are emerging trends.
    - Specialized algorithms for image analysis are being developed.
2. Video Search:
    - Challenges include identifying objects, actions, or events within videos.
    - Video analysis builds on image analysis but includes tracking objects across scenes and speech-to-text transcripts.
    - Techniques like video similarity browsing are being explored.
3. Audio Search:
    - Music-information retrieval systems allow users to query with musical content.
    - Systems like Shazam and Soundhound can recognize songs based on audio input.
    - Speech recognition technologies are emerging for identifying spoken words or phrases.
4. Geographic Information Search:

- Increasingly used to inform search queries, leveraging sensors and mobile devices.
- Map displays in user interfaces provide context for geographic queries.
- Challenges include dealing with spatial synonyms and user-generated geographic information.

5. Multilingual Searches:
   - Prototype systems allow searching multilingual collections of speech and printed documents.
   - Translation tools aid in accessing documents in languages users do not know.
6. Other Specialized Searches:
   - Designed for event sequences, graphs, structured document layouts, engineering diagrams, etc.
   - Specialized search interfaces cater to specific data types and domains.
   - Examples include graphical search interfaces for analyzing event sequences.

The Social Aspects of Search

Definition of Social Search:
- Involves search acts that utilize social interactions with others.
- Interactions can be explicit or implicit, synchronous or asynchronous, co-located or remote.
- Examples include searching for restaurant reviews on Yelp, asking for recommendations on social media platforms, and filtering results based on aggregated user interactions.

Explicit and Implicit Social Signals:
- Explicit signals include user ratings, reviews, and direct recommendations.
- Implicit signals include page rank, time spent on a page, mouse trails, and social media connections used by algorithms for result selection and suggestion.
- Personalization relies on building user profiles from interaction history for making personalized suggestions.

Collaborative Filtering and Recommender Systems:
- Groups of users combine their ratings to recommend items of interest to each other.
- Algorithms match users based on similar preferences and recommend items accordingly.
- Examples include movie recommendations on Netflix and product recommendations on Amazon.

Music Recommendation Systems:
- Platforms like Pandora and Last.fm combine personalization and collaborative filtering.
- User feedback and collaborative filtering generate playlists or track recommendations.
- Users provide feedback explicitly (likes) or implicitly (skips) to refine recommendations.

Challenges and Future Directions:

- Challenges include ensuring transparency in recommendation systems and enabling users to guide the recommendation process.
- Mobile personal assistants integrate personalized search and recommendations to anticipate user needs.
- Human-powered question answering platforms like Yahoo! Answers facilitate crowdsourced answers to user questions.
- Collaborative search, where users work together on search tasks, is an active research area with potential for future development in aiding collaborative efforts among users.

Ultimately, the timely user experience in searching is about providing efficient, effective, and understandable methods for users to find the information they need, minimizing frustration through good design, clear feedback, and appropriate response times. The computer acts as a "magic lens" to find, sort, filter, and present relevant items.

DATA VISUALIZATION

- Users deal with larger and more complex data in professional and personal contexts.
- Data representation medium affects ease of task performance.
- Visual representation is often preferred for conveying complex information.
- Visualization is defined as graphical representation of data to amplify cognition.
- Historical roots of visualization date back to pioneers like Playfair, Minard, Nightingale, and Snow.
- Visualization minimizes the gap between the user's mental model and system's state (Norman's gulfs of action).
- Visualization design is often empirical due to lack of specific macro-HCI theories.
- Computer-based visualization is interactive, offering opportunities beyond static representations.
- Effective interaction method for visualization minimizes gap of execution.
- Interaction for visualization differs from typical interfaces and user applications.
- Rapid evolution in computing, mobile technology, and data landscape since the visualization field's establishment.

TASKS IN DATA VISUALIZATION

- People interact with data for various reasons, and understanding these interactions is crucial for designing effective visualization tools.
- A pragmatic approach involves starting with the tasks users want to perform, which has been a focus of research in the visualization community for decades.
- The taxonomy consists of 12 task types grouped into three high-level categories:
    - data and view specification (visualize, filter, sort, derive),
    - view manipulation (select, navigate, coordinate, organize),
    - process and provenance (record, annotate, share, guide).
- These three categories incorporate the critical tasks that enable iterative visual analysis, including visualization creation, interactive querying, multi-view coordination, history, and collaboration.

Task Categories	*Task Types*
Data and view specification	**Visualize** data by choosing visual encodings
	Filter out data to focus on relevant items
	Sort items to expose patterns
	Derive values of models from source data
View manipulation	**Select** items to highlight, filter, or manipulate
	Navigate to examine high-level patterns and low-level detail
	Coordinate views for linked exploration
	Organize multiple windows and workspaces
Process and provenance	**Record** analysis histories for revisitation, review, and sharing
	Annotate patterns to document findings
	Share views and annotations to enable collaboration
	Guide users through analysis tasks or stories

1. Data and view specification

- Core functionality of any data visualization tool includes basic operations to visualize data using a visual representation, to filter out unrelated information, and to sort information to expose patterns.
- Users also need to derive new data from the input data, such as normalized values, statistical summaries, and aggregates.
- These four task types can be explained as follows:
    1. **Visualize data by choosing visual encodings:** This involves choosing the appropriate chart or visual representation for the dataset. Tools like Microsoft Excel and Tableau offer a palette of available charts, allowing users to pick the one most suitable for their data.
    2. **Filtering Data**: Filtering is essential for focusing on relevant items within a dataset. Various methods exist for filtering, such as lassoing important objects or using dynamic queries to select intervals and values on data dimensions. For example, in the context of a hotel search interface on a travel website like Kayak, users can dynamically query hotels based on filtering criteria like price intervals and features.
    3. **Sort items to expose patterns:** Ordering data items based on certain dimensions helps expose patterns within the data. Users can sort lists of items by clicking on the header category, with the option to toggle the order.
    4. **Derive values of 1nodels fro1n source data**: Original datasets can be augmented with derived data, such as statistics (e.g., mean, median), transformations, or data mining methods. Visual analytics, a research area that involves computational methods working in synergy with users, focuses on deriving insights from data through interactive systems.

2. View manipulation

Much of the value of visualization comes from being able to manipulate the view on the screen, including tl1e ability to select items or regions, to navigate the viewport's position on a large visualization, to coordinate multiple views so that data can be seen from multiple perspectives, and to organize the resulting dashboards and workspaces.

• ***Select items to highlight, filter, or manipulate.*** Pointing to an item or region of interest is common in everyday communication because it indicates the subject of conversation and action. In a visualization tool, common forms of selection include clicks (by mouse or by touch), mouse hover, and region selections (e.g., rectangular and elliptical regions or free-form lassos)

• ***Navigate to examine high-level patterns and low-level detail.*** Visualizations often contain dense information that can't all be displayed comfortably on screen. Navigation operations like pan and zoom allow users to control the viewport size and position within the visualization. Pan enables users to move around the visualization, while zoom adjusts the magnification level. These operations are common in various applications beyond visualization tools, such as Google Maps and Adobe Photoshop. Navigation tools are typically integrated into the user interface for easy access. Pan and zoom functionalities enhance the user experience by enabling effective exploration and interaction with visualizations.

• ***Coordinate views for linked exploration.*** Since each visualization technique has its own strengths and weaknesses, practical visualization tools often include multiple views of the same dataset so that each view illustrates a specific aspect of the data. When using multiple views in this way, such as in a visualization dashboard (Few, 2013), it is customary to coordinate the views so that selecting items in one view high lights the item (or related items) in other views (Fig. 16.3)

3. Process and provenance

- If the previous two categories of tasks deal with the mechanics of creating, manipulating, and viewing visualizations, the third category encompasses higher-level tasks for scaffolding, interpreting, and documenting the exploration process.
    1. ***Record analysis histories for revisitation, review, and sharing.*** Visualization tools do not only help users collect insights from their data, they should ideally also support mechanisms to record these insights as well as the path leading up to them. One approach that several tools provide is an automatically recorded history of interactions, allowing the user to review and revisit the exploration
    2. ***Annotate patterns to document findings.***Visualizations typically present data in a read-only format to aid user exploration. However, some tools allow users to add metadata through textual or graphical annotations. Textual annotations include labels, captions, or comments, while graphical annotations encompass sketches, highlights, or handwritten notes. To be truly useful, annotations should be data-aware, associated with underlying data points rather than overlaid on top of the visualization. This ensures their relevance and persistence even when the visualization is filtered or reorganized.
    3. ***Share views and annotations to enable collaboration*** Data analysis frequently entails collaboration among multiple users, necessitating visual analytics tools to include features for social interaction. These features range from basic

functionalities like exporting charts and datasets to advanced sharing mechanisms such as application bookmarking and web visualization publishing. By facilitating easy sharing, collaboration, and dissemination of insights, these tools empower users to engage in collaborative analysis within their professional networks or online communities.

4. ***Guide users through analysis tasks or stories*** Visualization tools are increasingly available to casual users exploring personal data. There's a rising need to guide novices in effective data analysis. Well-crafted data stories, combining visualizations, annotations, and textual descriptions, help explain complex phenomena succinctly.

CHALLENGES

Despite advancements and the emergence of frameworks like the task and data type taxonomies, data visualisation still faces numerous challenges for both researchers and practitioners:

Selecting the appropriate visualization technique: Choosing the most suitable technique for a given dataset and task remains a significant design problem.

Managing large volumes of data ("Big Data"): This is a general challenge as many tools, even commercial ones, struggle to maintain real-time interactivity with large datasets (only a few thousand or tens of thousands of items). Limitations are imposed by the number of available pixels on the screen and the inherent limits of human perception .

Evaluation: Evaluating complex data visualization systems is difficult. Challenges include:

Analysis is rarely an isolated, short-term process; tasks are often high-level

Users may need to look at the same data from different perspectives over long periods .

Users may formulate and answer questions they didn't anticipate, making typical empirical study techniques challenging

Significant discoveries, which can have a huge impact, occur rarely and are unlikely to be observed during a study

Case studies, while useful for reporting on users doing real tasks in natural environments, are very time-consuming and may not be replicable or applicable to other domains . Improvement in human-computer interaction requires measurement, and evaluation remains a challenge for visualisation researchers and practitioners

Data wrangling: Preparing the data before it can be visualized is a necessary prior step that presents challenges .

Integration of automatic algorithms: Continued integration of automatic algorithms with humans "in the loop" is needed to facilitate analytical reasoning. This research area is often called visual analytics

Social factors: As data visualization becomes more mainstream, social factors are increasingly important, including universal usability, supporting casual users, handling multiple device platforms, and focusing on dissemination through storytelling to lower entry barriers

Complexity of relationships and tasks: Visualizing complex data like networks remains an imperfect art, partly due to the complexity of the relationships within the data and the tasks users need to perform

Designing interface actions: Creating user interface actions that allow users to easily specify which relationships in visual displays should be manifested is still a challenge