Design of Interactive Systems (DIS)

Table of Contents

Lecture 1: Designing Interactive Systems

---

1. Overview of the Lecture

The first lecture in the course *Design of Interactive Systems (DIS)* lays the foundational principles for understanding how interactive systems are designed, evaluated, and optimized. The lecture is structured around multiple disciplines, including *Human-Computer Interaction (HCI)*, *User Experience (UX)*, and *Interaction Design (ID)*. It highlights essential concepts such as usability, user-centered design, and the role of technology in shaping user interactions.

The lecture is conducted by **Dr. Kalpana Shankhwar**, an **Assistant Professor at IIIT Delhi** with a background in **Mechanical Engineering**. Her research focuses on VR-based simulations, interactive wall development, gesture-based game design, and haptic devices for medical training.

---

2. What is an Interactive System?

An **interactive system** is a **technology-driven interface that allows users to interact with digital content in a meaningful way**. Examples of interactive systems include:

- **Smartphones**
- **Websites and Mobile Applications**
- **Automobiles with interactive dashboards**
- **Cash Dispensing Machines (ATMs)**
- **Interactive ticket vending machines**
- **Household smart appliances like smart refrigerators and voice assistants**

Characteristics of Interactive Systems

- They involve **information processing, transmission, storage, or transformation**.
- They should be **usable, accessible, and engaging**.
- They are built around human needs rather than technological capabilities.

A crucial aspect emphasized in the lecture is the **human-centered approach**, where **designing for users' needs, behaviors, and expectations** is a priority.

---

3. Key Disciplines in Interactive System Design

The lecture discusses three major disciplines that contribute to interactive system design:

A. Human-Computer Interaction (HCI)

> "HCI focuses on human-centeredness and usability concerns."

HCI ensures that technology is:

- **Easy to use**
- **Easy to learn**
- **Efficient in achieving user goals**

HCI encompasses methods, guidelines, and standards to ensure an **intuitive experience**.

B. Interaction Design (ID)

> “Interaction Design promotes a seamless interaction between users and products.”

Interaction Design is a sub-discipline within UX design that focuses on:

- **Microinteractions** (button clicks, animations, transitions)
- **Motion-based interactions** (scrolling effects, transitions)
- **Gestural inputs** (touch, swipe, pinch, voice)

Five Dimensions of Interaction Design

1. **1D - Words** (e.g., labels, instructions)
2. **2D - Visual Representation** (e.g., icons, colors)
3. **3D - Physical Objects/Space** (e.g., touchscreen, VR controls)
4. **4D - Time** (e.g., animations, loading time)
5. **5D - Behavior** (e.g., user response, AI predictions)

C. User Experience (UX) Design

> “UX design involves the entire process of acquiring and integrating a product, including branding, usability, and function.”

UX design is **holistic**—it considers:

- **User expectations and emotional responses**
- **Aesthetic and functional aspects of design**
- **Seamless integration of technology and human interaction**

Example: Apple’s UX Strategy

Apple’s product design (iPhone, MacBook) is a classic example of **good UX**. The intuitive interface, **minimalistic design**, **smooth animations**, and **accessibility features** make the user experience **effortless**.

---

4. The Role of UI Design

User Interface (UI) Design focuses on the **visual aesthetics and layout** of interactive systems. Key aspects include:

- **Color theory** for readability and engagement
- **Typography and font selection**
- **Iconography and symbols for easy navigation**

- **Layout structure (grid, alignment) for consistency**

UI design is crucial for **creating interfaces that are visually appealing and functionally efficient**.

Example: Google's UI Design Philosophy

- **Material Design Principles** focus on **real-world physics, clean typography, and intuitive motion**.
- **Dark Mode** in Android improves **accessibility and reduces eye strain**.

---

5. Importance of Being Human-Centered

A key takeaway from the lecture is that **design should prioritize human needs over technical features**. This is why:

1. **Users define success** – A product is only successful if users find it **usable and enjoyable**.
2. **Return on Investment (ROI)** – Good usability leads to **higher adoption rates, customer loyalty, and business growth**.
3. **Ethical Responsibility** – Interactive systems should be **inclusive** and **accessible to all users**, including those with disabilities.
4. **Sustainability** – Design choices should be **energy-efficient and environmentally friendly**.

Example: Accessibility in Design

Microsoft's **Windows Narrator and High Contrast Mode** allow visually impaired users to **navigate interfaces effectively**. Such features ensure **universal usability**.

---

6. Multimedia and Human Perception in Design

The lecture discusses how **human senses process information** in multimedia applications.

Human Perceptual System:

1. **Visual (input)**
2. **Acoustic (input/output)**
3. **Haptic (touch, skin sensors, motor system)**
4. **Taste & Smell (less relevant for digital interfaces)**

Multimodal Interactions in Interactive Systems

- **Voice-enabled assistants (Siri, Google Assistant)**
- **Haptic feedback in gaming (PS5 controllers with adaptive triggers)**
- **Gesture-controlled interfaces (Leap Motion, VR hand tracking)**

Example: Virtual Reality (VR)

> "VR creates highly-engaging user experiences."

VR **simulates real-world environments** using:

- **Head-mounted displays (Oculus, HTC Vive)**
- **3D spatial audio**
- **Hand tracking and motion sensing**

VR Applications

- **Medical Training** (haptic-based surgical simulations)
- **Education** (virtual classrooms)
- **Gaming** (immersive VR experiences)

---

7. Key Concerns in Interactive System Design

The lecture highlights the following concerns:

1. **What is Design?**
    - **Designing is problem-solving** through technology.
    - **User needs should define design choices**.
2. **Technology Integration**
    - Design should consider **emerging technologies like AI, IoT, and Blockchain**.
3. **Context and Activities**
    - Interactive systems should be **adaptable to different contexts**.
    - **Example:** A UI for a banking app should **prioritize security**, while a gaming UI should **emphasize engagement**.

---

8. Project and Evaluation Criteria

Projects in this course are meant to solve **real-world problems**. The evaluation criteria include:

- **Intermediate reports**
- **Final submission**
- **Code quality**
- **User testing and evaluation**
- **Presentations and demos**

Bonus Criteria

- **Novelty of the idea**
- **Presentation in workshops/conferences**
- **Real-time system implementations**

---

9. Class Activity

The class activity emphasizes:

1. **Identifying five interactive systems** used daily.
2. **Analyzing user experience (likes/dislikes).**
3. **Evaluating UI design elements.**

Example: Comparing UI of Spotify vs. Apple Music

- **Spotify**: More **personalized**, offers **algorithm-based recommendations**.
- **Apple Music**: More **structured**, offers **human-curated playlists**.
- **UX Takeaway**: Different UI approaches cater to different **user preferences**.

Conclusion

This first lecture on **Design of Interactive Systems** introduces the **principles of HCI, UX, and Interaction Design**, along with **technological considerations** in designing interactive experiences. It emphasizes the **human-centered approach**, ensuring usability, accessibility, and engagement.

Key Takeaways: ✔ Good design prioritizes user needs over technology.
✔ Interactive systems should be intuitive, accessible, and engaging.
✔ Emerging technologies (VR, AI, haptics) shape modern interaction paradigms.
✔ UX & UI Design principles ensure smooth user experiences.
✔ Projects should focus on real-world applications and problem-solving.

This lecture sets the foundation for deeper explorations in **usability, cognitive psychology, multimodal interactions, and emerging technologies** in interactive system design.

Lecture 2: PACT – A Framework for Designing Interactive Systems

---

1. Overview of the Lecture

Lecture 2 of *Design of Interactive Systems (DIS)* introduces **PACT**, a structured framework used in **human-centered interactive system design**. PACT stands for:

1. **People**
2. **Activities**
3. **Contexts**
4. **Technologies**

The **PACT framework** helps designers analyze **interactive systems** by considering the relationship between these four factors. By using PACT, designers ensure that systems are:

- **User-friendly**
- **Contextually relevant**
- **Efficient and accessible**
- **Technically feasible**

The lecture emphasizes the **human-centered approach**, where technology is **designed around human needs rather than forcing humans to adapt to technology**.

---

2. Understanding PACT: The Four Elements of Interaction Design

PACT is a **design thinking approach** that helps in structuring interactive system design by focusing on **users, their activities, the context in which they interact, and the technologies they use**.

A. People: The Human Factor

> "People differ in various ways, and design must consider these differences."

When designing an interactive system, it is crucial to understand that **no two users are alike**. People differ in:

- **Physical capabilities** (vision, hearing, dexterity)
- **Psychological traits** (memory, attention, cognitive abilities)
- **Social and cultural backgrounds** (language, habits, motivations)

1. Physical Differences

People experience the world through **five senses**:

- **Sight** (visual design, color contrast, accessibility)
- **Touch** (haptic feedback, touchscreens)
- **Hearing** (voice commands, audio cues)
- **Smell & Taste** (not common in digital design)

Example: Ergonomics in Design

> "The study of relationships between people and their environment."

- **Ergonomics** ensures that devices are **comfortable and safe** to use.
- Factors like **screen brightness, keyboard spacing, touch sensitivity** impact usability.
- Example: **Smartphone UI** is optimized for **one-handed use** by placing navigation controls at the bottom.

Fitts's Law

> "The time required to reach a target depends on its size and distance."

- If buttons are **too small or too far apart**, they become **difficult to use**.
- Example: **Large touch targets in mobile apps** improve accuracy.

2. Psychological Differences

- **Users remember concepts better than isolated details**.
- **Clear instructions and visual cues** improve usability.
- Some users prefer **text**, others prefer **images or videos**.

Mental Models in Design

> "Users form mental models of how a system works."

- If a **design aligns with user expectations**, learning is easier.
- Example: A **shopping cart icon** intuitively represents **adding items to a cart**.

Norman's Theory on Mental Models (1986)

- **Mental models are incomplete** – users may only understand parts of a system.
- **They are unstable** – users forget details over time.
- **They are unscientific** – users may develop superstitions about system behavior.
- Example: **Pressing the elevator button multiple times doesn't make it come faster, but users still do it**.

3. Social Differences

- Users have **different motivations** for using a system.
- **Beginners need guidance**, while **experts prefer shortcuts**.
- Some users **easily give up**, while others **explore features**.

Example: Netflix Personalization

- **Beginners** get **recommendations based on popularity**.
- **Long-time users** get **personalized content suggestions**.

B. Activities: Understanding User Interactions

Activities refer to **what users do with the system**. Different tasks require **different interaction designs**.

1. Temporal Aspects (Time-Based Factors)

- **Frequent vs. Infrequent use** – A **daily banking app** should have a simple login, while **a tax-filing app** can have a more detailed workflow.
- **Time pressure** – Systems used in **high-stress environments** (e.g., hospital software) should be **fast and error-free**.
- **Interruptions** – Users may be distracted, so **auto-save and undo features** are crucial.

2. Cooperation (Team-Based Workflows)

- Some activities involve **multiple users** (e.g., Google Docs allows real-time collaboration).

3. Complexity of Tasks

- **Simple tasks** (checking the weather) require **minimal UI**.
- **Complex tasks** (photo editing) require **advanced features but should remain user-friendly**.

4. Safety-Critical Tasks

- In fields like **aviation, healthcare, and finance**, **error prevention** is critical.
- Example: **Undo options in medical software prevent accidental deletions**.

Example: ATM User Experience

- **Simple, fast transactions** (cash withdrawal, balance check).
- **Clear, step-by-step instructions**.
- **Error prevention mechanisms** (e.g., timeout if the user forgets their card).

C. Context: The Environment in Which Interaction Occurs

Context refers to the **conditions under which users interact with the system**. The three major contexts are:

1. Organizational Context

- Corporate systems require **security, scalability, and efficiency**.
- Example: **Enterprise software like SAP and Salesforce**.

2. Social Context

- Some systems encourage **social interaction** (e.g., Facebook, WhatsApp).
- Example: **LinkedIn promotes professional networking**.

3. Physical Context

- Users might interact **indoors (home, office) or outdoors (while driving, at the gym)**.
- Example: **Voice commands in Google Assistant allow hands-free interaction**.

Example: ATM Design and Contextual Considerations

- Placed in **secure locations**.
- Screen visibility should be **optimized for outdoor use**.
- Should allow **multiple languages** for accessibility.

D. Technologies: Choosing the Right Tools

Technology refers to the **medium of interaction**. Designers must consider:

1. Input Technologies

- **Keyboards, touchscreens, voice recognition**.
- Example: **Touchless payment systems (Apple Pay, Google Pay)**.

2. Output Technologies

- **Displays (LCD, LED, AR/VR screens)**.
- **Audio feedback for accessibility**.

3. Communication Technologies

- **Wired vs. Wireless** (Wi-Fi, Bluetooth, 5G).
- Example: **Bluetooth earbuds allow hands-free communication**.

4. Content Considerations

- Content should be **relevant, up-to-date, and well-presented**.
- Example: **Streaming platforms optimize video quality based on internet speed**.

---

3. Scoping a Problem with PACT

A **PACT analysis** helps in:

- **Understanding user needs**.
- **Evaluating design decisions**.
- **Improving existing systems**.

Example: PACT Analysis for University Lab Access System

PACT Element	Analysis
People	Students, lecturers, technicians
Activities	Security clearance, opening the door
Contexts	Indoor, users carrying books
Technologies	Keycards, biometric scanners, PIN entry

---

4. Class Activity: PACT Analysis for a Vending Machine

The lecture ends with an activity where students must perform a **PACT analysis for a vending machine**.

Example Solution: PACT Analysis for Vending Machine

PACT Element	Analysis
People	Students, employees, visitors
Activities	Selecting an item, inserting money, retrieving the product
Contexts	Indoor/outdoor, noise levels, weather conditions
Technologies	Touchscreens, card payments, cash slots

Conclusion

PACT is a **systematic way** to **analyze and design** interactive systems by considering: ✔ **User diversity** (physical, psychological, social differences).
✔ **Task complexity** (frequent/infrequent tasks, time pressure).

✔ **Environmental context** (indoor/outdoor, individual/collaborative use).
✔ **Appropriate technologies** (input/output methods, content, communication).

By applying **PACT**, designers can create **human-centered, effective, and accessible interactive systems**.

Lecture 3: The Process of Human-Centered Interactive Systems Design

---

1. Overview of the Lecture

Lecture 3 in the *Design of Interactive Systems (DIS)* course introduces the **process of designing human-centered interactive systems**. The primary goal of this lecture is to establish a **systematic approach to designing interactive systems that prioritize human needs, behaviors, and usability**.

The lecture emphasizes:

1. **Understanding** – Researching user needs and requirements.
2. **Designing** – Developing conceptual and physical designs.
3. **Envisioning** – Creating representations of design ideas.
4. **Evaluating** – Testing and refining the design.

The lecture also introduces **scenarios and personas** as tools to understand user interactions and ensure the system aligns with their expectations.

---

2. The Core Principles of Human-Centered Interactive Systems Design

A. Design as a Creative and Iterative Process

> "Design is a creative process that involves conscious change and communication between designers and users."

Unlike traditional engineering, **interactive system design is not purely technical**; it involves an iterative, user-focused approach that continuously adapts based on evaluation and feedback.

Different disciplines approach design differently:

- Some designs are **stand-alone systems**.
- Others **integrate with existing (legacy) systems**.

Thus, a **flexible approach** is required to balance **creativity, usability, and technical feasibility**.

B. Four Key Activities in the Human-Centered Design Process

The **design process** involves four fundamental activities:

1. Understanding

> "Understanding is about knowing what the system has to do, what it has to be like, and how it fits into the ecosystem."

Designers must **study people, activities, and contexts** relevant to the domain. This step includes:

- **User research** – Gathering insights from real-world users.
- **Functional & Non-functional requirements**:
  - **Functional:** Core functionalities (e.g., an ATM should allow withdrawals).
  - **Non-functional:** Usability, performance, security.

Example: E-Commerce Website

- **Functional:** Users should be able to add products to the cart.
- **Non-functional:** The website should load in under 3 seconds.

2. Design

> "Design involves both conceptual design and physical design."

Design is **split into two stages**:

1. **Conceptual Design** – Abstract design process (focuses on 'what' rather than 'how').
   - Example: **Use Cases**, **Entity-Relationship Models**, **Wireframes**.
   - Keeps things **abstract** to define **core interactions** before implementation.
2. **Physical Design** – Making abstract concepts **concrete**.
   - Focuses on **interface elements, interactions, and system architecture**.

Example: Banking App

- **Conceptual Design:** Sketching the flow of login, balance check, and transactions.
- **Physical Design:** Defining UI components (buttons, color schemes, typography).

3. Envisionment

> "Designs need to be visualized to clarify ideas and enable evaluation."

Before building a full system, **prototypes and mock-ups** are created:

- **Low-Fidelity Prototypes** – Sketches, storyboards, paper prototypes.
- **High-Fidelity Prototypes** – Functional prototypes with limited features.

Example: Mobile App Design

- **Sketches** of different screens help refine the layout.
- **Interactive mock-ups** allow testing before full implementation.

4. Evaluation

> "Evaluation ensures the design meets user needs and functions correctly."

Evaluation is **continuous** throughout the process. It includes:

- **Designer self-checks** (verifying requirements).
- **Client feedback** (reviewing wireframes and prototypes).
- **User testing** (observing real users interact with the system).

Example: A/B Testing for a Website

- Two designs are tested on real users.
- The one with **higher engagement** and **better usability** is chosen.

---

3. Developing Personas and Scenarios

Personas and scenarios are essential for human-centered design.

A. Personas: Representing Users

> "Personas are concrete representations of the different types of people who will use the system."

A persona should include:

- **Name & Background**
- **Goals & Aspirations**
- **Pain Points (Problems They Face)**
- **Technology Comfort Level**

Example: Persona for a Fitness App

Attribute	Details
Name	Ayesha Sharma
Age	28
Occupation	Software Engineer
Goals	Wants to track calories and exercise progress easily
Pain Points	Finds most fitness apps too complex
Tech Skills	Intermediate

Using this persona, the app should have: ✔ A **simple UI** with clear calorie tracking.
✔ **Easy logging of workouts** without complex steps.

B. Scenarios: Real-World User Interactions

> "Scenarios are stories about users engaging with the system in different contexts."

Scenarios help designers **visualize user interactions**. They are **progressively detailed**:

1. **Stories** – Real-world experiences (e.g., diary entries, observations).
2. **Conceptual Scenarios** – Abstract descriptions removing unnecessary details.
3. **Concrete Scenarios** – Adding specific design elements and interactions.
4. **Use Cases** – Defining **detailed system interactions**.

Example: ATM Use Case

Scenario	Details
Story	A user needs to withdraw cash.
Conceptual Scenario	A person interacts with an ATM for a transaction.
Concrete Scenario	User enters PIN, selects amount, confirms withdrawal.
Use Case	The ATM system verifies credentials, deducts balance, and dispenses cash.

---

4. Scenario-Based Design Method

The **Scenario-Based Design Method** formalizes how **different scenarios help in different design stages**:

1. **Understanding Users** – Stories capture real-world insights.
2. **Envisioning Designs** – Conceptual scenarios help designers brainstorm.
3. **Prototyping and Testing** – Concrete scenarios define the system structure.
4. **Final Implementation** – Use cases refine user interactions and system logic.

---

5. Implementation and Product Development

> "Ultimately, systems must be implemented, tested, and launched."

Once the design process is complete: ✔ The system is **developed and tested**.
✔ **Bugs and issues are fixed** before launch.
✔ The **final product is deployed** to users.

Example: Developing a Smart Home App

1. **Conceptual Design** – Outlining how users will control lights and appliances.
2. **Physical Design** – Designing the UI for mobile and voice control.
3. **Envisionment** – Creating interactive prototypes.
4. **Evaluation** – User testing and feedback.
5. **Implementation** – Coding and integrating with smart devices.

---

6. Class Activity

Task: Analyze a vending machine system using personas and scenarios.

1. **Observe** how users interact with vending machines.

2. **Document real-life stories** (e.g., a user struggling with payment).
3. **Develop a conceptual scenario** (e.g., user selects a product and makes payment).
4. **Define a concrete scenario** (e.g., user inserts a card, machine verifies, dispenses product).
5. **Create a use case** (e.g., system logic for product selection, payment processing, and dispensing).

Conclusion

This lecture provides a **structured approach to human-centered interactive system design**, emphasizing:
✔ **Understanding user needs and requirements**.
✔ **Conceptual and physical design principles**.
✔ **The role of personas and scenarios in interaction design**.
✔ **The iterative process of envisionment, evaluation, and implementation**.

By following this approach, designers can create **intuitive, efficient, and user-friendly interactive systems**.

Lecture 4: Usability in Interactive Systems

---

1. Overview of the Lecture

Lecture 4 of *Design of Interactive Systems (DIS)* focuses on **Usability**, which is central to **Human-Computer Interaction (HCI)**. The lecture defines **usability**, explores **key usability principles**, and examines how **design principles** impact user experience.

The lecture emphasizes:

- **Usability Goals** – How to make systems **easy to use, efficient, and flexible**.
- **Accessibility and Acceptability** – Ensuring systems are **usable by diverse users**.
- **Norman's Usability Model** – The **gulf of execution** and **gulf of evaluation**.
- **Key Design Principles** – Including **visibility, consistency, feedback, and error recovery**.

---

2. What is Usability?

> *"Usability is that systems should be easy to use, easy to learn, flexible, and should engender a good attitude in people (Shackel, 1990)."*

Usability is the **quality of user interaction** with a system. It ensures that **users can efficiently and effectively complete tasks without frustration**.

A system must be:

1. **Accessible** – No barriers to usage.
2. **Usable** – Minimal effort required to achieve goals.
3. **Acceptable** – Fit for purpose in real-world contexts.

---

3. The Three Views of Good Design

Good design is subjective and depends on context. However, interactive systems designers typically follow **three main perspectives**:

1. **View 1:** Systems should be **accessible, usable, socially, and economically acceptable**.
2. **View 2:** Systems should be **learnable, effective, and accommodating** to diverse users.
3. **View 3:** The **PACT framework (People, Activities, Contexts, Technologies)** should be balanced.

These perspectives ensure that **interactive systems are practical, efficient, and inclusive**.

---

4. Accessibility in Interactive Systems

> *"A system must be accessible before it is usable."*

Accessibility ensures **no user is excluded from using a system**. **Legal regulations**, such as the **UK's Equality Act 2010** and **Section 508 in the USA**, mandate **software accessibility**.

Barriers to Accessibility

Users can be excluded due to:

1. **Physical Barriers** – Limited mobility, visual impairments.
2. **Conceptual Barriers** – Users failing to develop a mental model of the system.
3. **Economic Barriers** – High costs preventing access.
4. **Cultural Barriers** – Poor localization, misunderstanding of cultural norms.
5. **Social Barriers** – Systems being unavailable at convenient times or places.

Overcoming Accessibility Barriers

There are **two key approaches**:

- **Universal Design ("Design for All")** – Systems should be **usable by everyone** from the outset.
- **Inclusive Design** – Designs should accommodate a **broad range of abilities**.

Example: Accessibility in Modern Systems

✔ **Voice assistants (Siri, Google Assistant)** support users with limited mobility.
✔ **Screen readers (NVDA, JAWS)** help visually impaired users navigate digital content.
✔ **High-contrast modes** improve readability for users with low vision.

Key takeaway: *"If a design works well for people with disabilities, it works better for everyone."*

---

5. What Makes a System Usable?

A system with **high usability** possesses the following characteristics:

1. **Efficiency** – Users complete tasks with **minimum effort**.
2. **Effectiveness** – The system provides **relevant functions and content**.
3. **Learnability** – Users can **quickly grasp how to use** the system.
4. **Safety** – The system **prevents major errors and failures**.

5. **Utility** – The system does what users **expect and need**.

---

6. Four Key Principles of Usability

Usability is **not an afterthought**—it must be **embedded into the design process**. The following principles ensure usability is maintained:

1. Early Focus on Users and Tasks

- Designers must **study user needs and behaviors**.
- **Participative Design:** Users should be involved in **design decisions**.

Example:
✔ **Amazon's website redesigns** involve extensive **user research and A/B testing**.

2. Empirical Measurement

- Usability should be **measured using real user data**.
- **Testing methods:** User feedback, usability tests, eye-tracking studies.

Example:
✔ **Google's search algorithm updates** are based on **user interaction data**.

3. Iterative Design

- Design must be **continuously tested and refined**.
- Designers must **fix usability problems** through cycles of **testing and improvement**.

Example:
✔ **Windows 11 UI refinements** came from **iterative user testing and feedback**.

4. Integrated Usability

- Usability must be **holistic**—**all elements of the system** should evolve together.

Example:
✔ **Apple's ecosystem** (iPhone, iPad, Mac) provides a **seamless, integrated experience**.

---

7. Norman's Usability Model: Bridging the Two Gulfs

> *"Technology should not get in the way of what people want to do."* – **Don Norman (1988)**

Norman identifies two **"gulfs"** in usability:

1. **Gulf of Execution** – Users struggle to **translate their goals into system actions**.
    - Example: A **TV remote with too many buttons** confuses users.
2. **Gulf of Evaluation** – Users struggle to **determine if their actions were successful**.
    - Example: **Unclear error messages in software** leave users frustrated.

Bridging the Gulfs

To **reduce usability gaps**, designers should: ✔ Use **intuitive icons and labels**.
✔ Provide **clear feedback** for user actions.
✔ Ensure **interfaces match user expectations**.

Example:
✔ **Google Maps** automatically **reroutes users** when they make a wrong turn, **bridging the Gulf of Evaluation**.

---

8. Acceptability in Interactive Systems

> *"Acceptability is about fitting technology into people's lives."*

Key Factors of Acceptability

1. **Political** – Systems must **comply with laws and ethical standards**.
2. **Convenience** – Users should **find the system easy to integrate** into their routines.
3. **Cultural & Social** – The system must be **sensitive to cultural differences**.
4. **Usefulness** – The system should **solve real-world problems**.
5. **Economic** – Users must **afford and justify** the system's cost.

Example:
✔ **Digital wallets (Paytm, Google Pay)** are widely adopted due to their **convenience and economic benefits**.

---

9. 12 Key Design Principles for Usability

Usability principles ensure systems are **intuitive, efficient, and error-free**.

Learnability: Helping Users Access, Learn, and Remember

1. **Visibility** – Users should see available options **clearly**.
2. **Consistency** – Interfaces should follow **established patterns**.
3. **Familiarity** – Use **common symbols and terminology**.
4. **Affordance** – Make elements **obviously interactive**.

Effectiveness: Ensuring Smooth Interaction

5. **Navigation** – Provide **clear directions and wayfinding cues**.
6. **Control** – Users should feel in **control** of the system.
7. **Feedback** – Systems should provide **immediate responses** to actions.

Safety: Minimizing Errors and Risks

8. **Recovery** – Allow **undo options and error handling**.
9. **Constraints** – Prevent users from **making invalid actions**.

Accommodation: Supporting Different User Needs

10. **Flexibility** – Provide **multiple ways to complete tasks**.
11. **Style** – Aesthetics should enhance, not distract from usability.
12. **Conviviality** – Interfaces should be **friendly and polite**.

---

10. Class Activity

Students are asked to **identify interactive systems** that align with each **usability principle**.

Example: Applying Principles to Everyday Systems

Design Principle	Example System
Visibility	Google Search Bar
Consistency	Microsoft Office UI
Navigation	GPS Navigation Systems
Feedback	Mobile Banking Apps

Conclusion

✔ **Usability is essential for ensuring smooth user interaction.**
✔ **Accessibility and inclusivity improve overall user experience.**
✔ **Norman's model helps designers reduce usability gaps.**
✔ **Key usability principles guide iterative, user-centered design.**

By following these principles, **interactive systems can be made efficient, accessible, and enjoyable for all users**.

Lecture 5: Experience Design in Interactive Systems

---

1. Overview of the Lecture

Lecture 5 of *Design of Interactive Systems (DIS)* introduces **Experience Design**, an essential aspect of **Human-Computer Interaction (HCI)**. Unlike traditional usability-focused design, **experience design aims to create enjoyable, engaging, and memorable interactions** for users.

The lecture explores:

- **User Experience (UX)** – Beyond usability, focusing on **emotions and engagement**.
- **Nathan Shedroff's Model** – Key elements of **identity, adaptivity, narrative, immersion, and flow**.
- **Gamification & Fun Models** – How **fun and emotions** drive engagement.
- **Designing for Pleasure** – The role of **aesthetics, emotional responses, and product attachment**.

Experience design ensures that interactive products are **not just functional but delightful to use**.

---

2. What is Experience Design?

> *"Designers of interactive systems are increasingly expected to design systems that provide great experiences."*

Experience Design (XD) is about creating products that evoke **emotion, engagement, and satisfaction**.

For example: ✔ A **shopping list app** should not just be **functional**—it should be **fun to use**.
✔ A **website** should not just display information—it should **keep users engaged**.

> *"UX design goes beyond usability—it's about creating interactions that are immersive, engaging, and meaningful."*

Aims of Experience Design

1. Explore **different traditions** influencing experience design.
2. Understand **Nathan Shedroff's Model** of engagement.
3. Learn **how aesthetics impact experience**.
4. Design for **pleasure, immersion, and long-term attachment**.

---

3. Key Factors in Experience Design

Experience design is **concerned with all the qualities that make an interaction memorable, satisfying, and enjoyable**.

Examples of engaging experiences:

- **A good book** pulls the reader into its story.
- **A video game** provides an immersive challenge.
- **A well-designed app** makes users feel in control.

Emotion & Experience

> *"Experience is about feeling."*

- Emotion is **central to experience design**.
- **Users don't just use a system—they feel something while interacting with it**.
- **Designers can't create experiences directly**—they can only design **for** experiences.

Example: Why Apple's UX is successful ✔ **MacBooks** feel premium and aesthetically pleasing.
✔ **iPhones** create a **sense of luxury and exclusivity**.
✔ **Animations & gestures** provide **smooth, satisfying interactions**.

---

4. Nathan Shedroff's Model of Engagement

> *"Engagement is about ensuring that interaction flows smoothly."*

Engagement occurs when all **PACT elements** (People, Activities, Contexts, and Technologies) **harmonize**.

Nathan Shedroff identifies 5 key elements of engagement:

Element	Description	Example

Element	Description	Example
Identity	Users feel connected to the system, developing a sense of ownership.	Apple vs. Windows users
Adaptivity	The system adapts to users, offering personalization.	Netflix's AI-driven recommendations
Narrative	A compelling story structure keeps users engaged.	Instagram Stories, video games
Immersion	Users feel completely involved in the experience.	VR gaming, interactive learning
Flow	Smooth transitions and intuitive actions make interactions seamless.	Swiping gestures on a smartphone

---

5. Gamification & Fun in Interactive Systems

> *"Games engage players by triggering emotions—curiosity, excitement, amusement, and satisfaction."*

A. The Four Fun Keys (Lazzaro, 2012)

Lazzaro identified **four key types of fun** in interactive experiences:

Fun Type	Key Emotion	Example
Hard Fun	**Fiero (Triumph over challenge)**	Competitive games like *Dark Souls*
Easy Fun	**Curiosity & Exploration**	Open-world games like *Minecraft*
Serious Fun	**Relaxation & Mastery**	Meditation apps, educational games
People Fun	**Social Interaction & Amusement**	Multiplayer games like *Among Us*

B. How Emotions Enhance Engagement

Lazzaro identified **five ways emotions enhance gaming experiences**:

1. **Enjoyment** – Strong internal emotional shifts.
2. **Focus** – Helps players concentrate.
3. **Decision-making** – Emotions influence choices.
4. **Performance** – Engagement boosts performance.
5. **Learning** – Emotions **improve motivation and retention**.

✔ **Example:** *Super Mario* keeps players engaged by balancing **challenge, rewards, and surprises**.

---

6. Designing for Pleasure

> *"Pleasure is as important as usability."*

Donald Norman (2004) argues that designs should evoke pleasure—not just serve a function.

Four Dimensions of Pleasure (Tiger, 1992)

Pleasure Type	Description	Example
Physio-Pleasure	Sensory appeal (touch, texture, sound)	The smooth finish of an iPhone
Socio-Pleasure	Social connection & status	Owning a luxury watch
Psycho-Pleasure	Cognitive satisfaction (ease of use)	A well-organized task management app
Ideo-Pleasure	Aligning with personal values	Buying **eco-friendly** products

✔ **Example:** *MacBook Air Analysis*

- **Physio-Pleasure**: Lightweight, responsive keyboard.
- **Socio-Pleasure**: Symbol of tech-savvy individuals.
- **Psycho-Pleasure**: Simple, user-friendly interface.
- **Ideo-Pleasure**: Associated with creativity & innovation.

---

7. Product Attachment: Why Users Stay Loyal

> *"Users form emotional attachments to products based on their personal identity and experiences."*

Researchers identify **six framing constructs** behind product attachment:

Construct	Description
Role Engagement	Supports users' roles (e.g., student, professional, gamer)
Control	Users want customization (e.g., skins, themes)
Affiliation	The product becomes a **part of social identity**
Ability & Habit	Enhances user abilities & avoids bad habits
Long-term Goals	Supports lifelong learning & growth
Ritual	Fits into users' daily routines

✔ **Example:** *Smartwatches like Apple Watch*

- Helps with **fitness tracking** (Role Engagement).
- Allows **customization** (Control).
- Signals **status** (Affiliation).

---

8. Aesthetics in Experience Design

> *"Aesthetics influence how people feel about a product before they even use it."*

Donald Norman's Three Levels of Aesthetics (2004)

1. **Visceral Design** – Immediate, instinctive response (e.g., "This looks cool!").
2. **Behavioral Design** – Satisfaction from usability & functionality.

3. **Reflective Design** – Personal meaning & identity (e.g., "This device represents me!").

✔ **Example:** *Why People Love Tesla Cars*

- **Visceral**: Sleek design, minimal dashboard.
- **Behavioral**: Smooth acceleration, self-driving features.
- **Reflective**: Owning a Tesla = Environmental consciousness.

---

9. Measuring Product Emotions

Designers use **PrEmo (Product Emotion Navigator)** to evaluate emotional responses:

14 Core Emotions in Product Design

Positive Emotions	Negative Emotions
Inspiration	Disgust
Desire	Indignation (Anger)
Satisfaction	Contempt
Pleasant Surprise	Disappointment
Fascination	Dissatisfaction
Amusement	Boredom

✔ **Example:** *Why people love iPhones*

- **Satisfaction** (Smooth UI).
- **Pleasant Surprise** (New animations).
- **Fascination** (Premium feel & branding).

---

10. Class Activities

1. **Analyze your favorite game** using the **Four Fun Keys**.
2. **Describe a product you are emotionally attached to** using **six framing constructs**.

Conclusion

✔ **Experience design goes beyond usability—engagement and emotions matter.**
✔ **Gamification makes interactions more fun and rewarding.**
✔ **Pleasure and aesthetics shape long-term product attachment.**
✔ **Emotions influence user satisfaction, loyalty, and overall experience.**

By designing **for** experience, **interactive systems become not just functional, but delightful to use**.

Lecture 6: Techniques for Designing Interactive Systems – Understanding, Envisionment, and Design

---

1. Overview of the Lecture

Lecture 6 of *Design of Interactive Systems (DIS)* explores **techniques for designing interactive systems**, focusing on:

- **Understanding** – Gathering user requirements through research.
- **Envisionment** – Externalizing design ideas through sketches, prototypes, and wireframes.
- **Design** – Creating conceptual and physical designs.

The lecture emphasizes **user-centered design**, ensuring that interactive systems **meet real-world needs** effectively.

---

2. Understanding User Needs in Interactive System Design

> *"Before creative design can start, the designer must develop a clear understanding of PACT."*

What is "Understanding" in Design?

Understanding refers to **researching and analyzing**:

1. **People** – Who will use the system?
2. **Activities** – What will users do with the system?
3. **Contexts** – Where and how will they interact with it?
4. **Technologies** – What tools will be used?

This step helps **define system requirements**.

Understanding Requirements

A **requirement** is:

> *"Something the product must do or a quality the product must have."*

Example: Online Banking App ✔ **Functional Requirements** – Users should log in, check balances, and transfer money.
✔ **Non-Functional Requirements** – The app must be **secure**, **fast**, and **mobile-friendly**.

Prioritizing Requirements – MoSCoW Rules

A method for sorting requirements into priority levels:

Category	Definition	Example (E-commerce Website)
Must Have	Critical for the system to work	Secure login, checkout process

Category	Definition	Example (E-commerce Website)
Should Have	Important but not essential	Wishlist feature, product comparison
Could Have	Nice to have, adds value	AI-based recommendations
Won't Have (for now)	Not needed in this version	Augmented Reality (AR) preview

This approach ensures that **critical functions are implemented first**.

---

3. Techniques for Gathering User Requirements

To **understand user needs**, designers use multiple research methods.

A. Participative Design

> *"Designers must understand the needs of users by involving them in the design process."*

Techniques include: ✔ **Interviews** – Talking directly to stakeholders. ✔ **Observations** – Watching users interact with existing systems. ✔ **Workshops & Focus Groups** – Gathering feedback in group discussions.

B. Interviews

> *"One of the most effective ways to understand user needs."*

Types of Interviews:

1. **Structured Interviews** – Pre-defined, fixed questions.
2. **Semi-Structured Interviews** – Guided but flexible.
3. **Unstructured Interviews** – Open-ended, exploratory.

✔ **Example:** Interviewing bank customers about their experience using online banking.

C. Questionnaires

> *"Useful for large-scale surveys when individual interviews aren't practical."*

Good questionnaires: ✔ Are **clear and unambiguous**.
✔ Gather **relevant data**.
✔ Are **easy to analyze**.

Example: Conducting an online survey about **food delivery app usability**.

D. Observing Users in Their Environment (Fieldwork)

> *"People may not always explain their behavior accurately—observing them is key."*

✔ **Example:** Watching users interact with **ticket vending machines** in a subway.

✔ **Advantage: Realistic data** about how users behave in **actual settings**.

---

4. Envisionment – Visualizing Design Ideas

> *"Envisionment is about externalizing thoughts—making ideas visible."*

Once user requirements are clear, designers **visualize** solutions through: ✔ **Sketches & Snapshots** ✔ **Storyboards** ✔ **Moodboards** ✔ **Navigation Maps** ✔ **Wireframes** ✔ **Prototypes**

A. Finding Suitable Representations

> *"Different representations are used at different design stages."*

✔ **Example: Designing a Sports Car**

- **Doodles & Sketches** – Early brainstorming.
- **Blueprints & Scale Models** – Refining the design.
- **Wind Tunnel Testing** – Evaluating aerodynamics.
- **Computer Models** – Predicting real-world performance.

B. Steps in the Envisionment Process

1. **Review requirements** and conceptual scenarios.
2. **Develop representations** (sketches, wireframes, prototypes).
3. **Explore different design metaphors**.
4. **Test ideas with users**.
5. **Iterate and refine designs**.

This process **ensures designs align with user needs**.

C. Envisionment Techniques

Technique	Description	Example
Sketches & Snapshots	Quick hand-drawn ideas	UI layout sketches
Storyboards	Sequence of images showing interactions	Step-by-step checkout process
Moodboards	Visual inspiration, colors, fonts	Branding design
Navigation Maps	User journey through a system	Website flowchart
Wireframes	Structural layout without visuals	Low-fidelity app design
Prototypes	Functional but incomplete models	Clickable app demo

✔ **Example: Storyboarding a Music Streaming App**

1. **User opens the app** → 2. **Searches for a song** → 3. **Plays the song** → 4. **Adds it to a playlist**.

---

5. Design – Conceptual vs. Physical

> *"Design is about structuring interactions into logical sequences and refining the look and feel of a product."*

A. Conceptual Design

- **Abstract** – Focuses on **logic and structure**.
- Defines **system functions, content, and workflow**.

✔ **Example:** Designing an **ATM user flow** (insert card → enter PIN → withdraw cash).

B. Physical Design

- **Concrete** – Focuses on **interface, interaction, and aesthetics**.
- Defines **layout, colors, fonts, animations, and physical components**.

✔ **Example:** Choosing the **button layout on an ATM touchscreen**.

---

6. Key Design Concepts

A. Exploring Design Space

> *"Design constraints help focus creativity while allowing flexibility."*

✔ **Example:** Large font size in mobile apps:

- **Pros:** Improves readability.
- **Cons:** Takes up screen space.

B. Metaphors in Design

> *"Using familiar concepts from one domain to explain another."*

✔ **Example: "Shopping cart" in e-commerce**—users intuitively understand it.

---

7. Physical Design and Interaction

A. Objects & Actions

✔ **Example:** MP3 Player Design

- **Object:** Play Button
- **Action:** Press to start music

B. Design Languages

A **design language** consists of:

1. **Design Elements** – Colors, fonts, buttons, sliders.
2. **Composition Rules** – How elements are arranged.
3. **Contextual Guidelines** – Adjusting designs for different devices.

✔ **Example: Material Design (Google)** follows strict guidelines for UI components.

---

8. Class Activity

Scenario: Designing a **16A heavy-duty plug socket**.

Steps:

1. **Create a scenario corpus** (stories to use cases).
2. **Identify key requirements** (safety, ease of use, durability).
3. **Propose a prototype concept** (3D model, interactive UI).

✔ **Example: Improving Industrial Power Sockets**

- **Design Challenge:** Users struggle with plugging/unplugging in **low-light environments**.
- **Solution:** Glow-in-the-dark indicators + ergonomic grip.

---

9. Conclusion

✔ **Understanding user needs is the foundation of good design.**
✔ **Research methods (interviews, surveys, observations) improve requirement gathering.**
✔ **Envisionment techniques (storyboards, wireframes, prototypes) bring ideas to life.**
✔ **Conceptual and physical design shape the system's logic and appearance.**
✔ **A consistent design language ensures usability and brand recognition.**

By applying these techniques, **interactive systems can be user-friendly, efficient, and engaging**.

Lecture 7: Techniques for Evaluating Interactive Systems

---

1. Overview of the Lecture

Lecture 7 of *Design of Interactive Systems (DIS)* covers **evaluation techniques** used to **assess interactive systems**. Evaluation ensures that **designs are usable, effective, and engaging** before full implementation.

The lecture focuses on: ✔ **Expert-based evaluation** – Usability experts review designs based on established principles.
✔ **Participant-based evaluation** – Real users test the system to identify **practical usability issues**.
✔ **Evaluation metrics** – Methods for measuring usability, engagement, and user satisfaction.

Evaluation **identifies usability problems early**, reducing costs and ensuring a **better user experience**.

---

2. What is Evaluation in Interactive System Design?

> *"Evaluation is the process of reviewing and testing a design idea, piece of software, product, or service."*

Key Goals of Evaluation

1. **Assess usability** – How easy is the system to learn and use?
2. **Check effectiveness** – Does it perform its intended functions well?
3. **Measure engagement** – Is the experience enjoyable and immersive?
4. **Ensure accessibility** – Can diverse users interact with it?

✔ **Example:** Evaluating an **e-commerce website**

- **Usability:** Are checkout steps simple and fast?
- **Effectiveness:** Do search filters work correctly?
- **Engagement:** Is the UI visually appealing?
- **Accessibility:** Does it support screen readers for visually impaired users?

---

3. Challenges in Evaluation

Evaluating **different systems and contexts** presents unique challenges.

Key Evaluation Challenges

1. **Different types of systems** – Evaluating a **mobile app** differs from evaluating **VR applications**.
2. **Context variability** – User behavior **changes** based on **environment and task complexity**.
3. **Diverse users** – Different **experience levels** and **abilities** must be considered.
4. **Evaluation timing** – Early-stage evaluation may use **prototypes**, while late-stage evaluation involves **fully functional systems**.

✔ **Example:** Evaluating a **smart home assistant**

- Users interact **differently at home vs. in a lab setting**.
- Users **expect natural voice commands** (which must be tested in real-world settings).

---

4. Types of Evaluation

Evaluation is classified into **two major types**:

A. Expert-Based Evaluation

> *"A usability expert or interaction designer evaluates the system without real users."*

✔ **Faster & cost-effective**, but may **miss real-world user frustrations**.

B. Participant-Based Evaluation

> *"Real users interact with the system to identify usability issues."*

✔ **Captures real user behavior**, but **requires more time and resources**.

✔ **Example:** Testing a **fitness tracking app**

- **Expert Evaluation:** A designer reviews whether menus are intuitive.

- **Participant Evaluation:** Users test whether the app **correctly tracks workouts**.

---

5. Expert-Based Evaluation Methods

> *"Experts analyze the system based on usability principles and known design patterns."*

A. Heuristic Evaluation

> *"A usability expert checks if the design follows established heuristics."*

✔ Based on design principles (heuristics):

Heuristic Principle	Explanation	Example
1. Visibility	Users should see available options clearly.	Large, well-labeled buttons.
2. Consistency	UI elements should behave predictably.	Uniform icons in a mobile app.
3. Familiarity	Use familiar conventions.	A shopping cart icon for e-commerce.
4. Affordance	Design should indicate function.	A button should look clickable.
5. Navigation	Users should move smoothly.	Breadcrumb navigation in websites.
6. Control	Users should feel in charge.	Undo and redo options.
7. Feedback	Immediate response to user actions.	"Item added to cart" confirmation.
8. Recovery	Easy correction of errors.	"Forgot password?" option in login screens.
9. Constraints	Prevent invalid inputs.	Only allowing numbers in a phone number field.
10. Flexibility	Support different user skill levels.	Keyboard shortcuts for power users.
11. Style	Visually appealing design.	Clean, aesthetic UI.
12. Conviviality	Pleasant and user-friendly interaction.	Personalized greetings in apps.

✔ Example: Gmail's Heuristic Evaluation

- **Good:** Consistent interface across devices.
- **Issue:** Finding old emails can be complex.
- **Solution:** Improved search filters and AI-powered suggestions.

B. Cognitive Walkthrough

> *"Evaluates step-by-step interaction to detect usability issues."*

Key Questions:

1. Will users know what to do?
2. Will users find the right action?
3. Will they associate the action with their goal?
4. Will users see that they made progress?

✔ **Example: Evaluating a Banking App**

- Users must **transfer money**.
- The walkthrough **checks if users can complete the task smoothly**.

If users **struggle at any step**, **usability issues** need fixing.

C. Discount Usability Engineering

> *"A quick, low-cost usability review based on three core principles."*

Principle	Description
Learnability	How quickly users understand the interface.
Effectiveness	How well users complete tasks.
Accommodation	Whether the system adapts to user needs.

✔ **Example:** Evaluating a **mobile ticket booking app** by checking:

- How **quickly** new users learn to book tickets.
- Whether the app **prevents booking mistakes**.

---

6. Participant-Based Evaluation Methods

> *"Users interact with the system while researchers observe their experience."*

A. Cooperative Evaluation

> *"Users work as co-evaluators, giving real-time feedback."*

✔ **Example:** Users test a **new video streaming app**, providing feedback as they navigate.

B. Participatory Heuristic Evaluation

> *"Users and experts evaluate together."*

✔ **Example:** Designers and users **jointly review** a **navigation system** for self-driving cars.

C. Co-Discovery

> *"Two users explore the system together and discuss their thoughts."*

✔ **Example:** Two **first-time users** test a **health tracking app**, discussing confusion points.

D. Controlled Experiments

> *"Comparing two versions of a design to see which performs better."*

✔ **Example:** A/B testing two **checkout page designs** in an **e-commerce store**.

---

7. Metrics and Measures in Evaluation

✔ **Objective usability metrics ensure accurate evaluation.**

Metric	Definition	Example
Time to complete a task	How long users take to finish an action.	How long to book a flight.
Error rate	How often users make mistakes.	Incorrect form submissions.
Success rate	Percentage of users completing a task successfully.	Percentage of users completing checkout.
Satisfaction score	User ratings of their experience.	App Store ratings and feedback.

✔ **Example:** Measuring **VR app usability**

- **Metric:** Users' **reaction time** when interacting with virtual objects.

---

8. Reporting Usability Evaluation Results

After evaluation, findings must be **documented and reported**.

✔ **Key Reporting Elements:**

- **List of issues found.**
- **Severity of each issue.**
- **Proposed solutions.**

✔ **Example: Evaluating a smart fridge UI**

- **Issue:** Users struggle to find the temperature control.
- **Solution:** Make controls **more prominent**.

---

9. Advanced Evaluation Techniques

✔ **Eye-tracking** – Measures **where users focus on a screen**.
✔ **Physiological Measures** – **Heart rate, skin response** track emotional reactions.
✔ **Evaluating "Presence" in VR** – Measures how immersive the experience feels.

✔ **Example:** Evaluating **haptic feedback in VR shopping**

- Users **feel and interact** with products virtually.
- Evaluation **measures engagement & usability**.

Conclusion

✔ **Evaluation ensures systems are usable, effective, and engaging.**
✔ **Expert evaluations identify early design flaws.**
✔ **User testing captures real-world behavior.**
✔ **Metrics and data-driven insights improve designs.**

By applying these techniques, **interactive systems can be optimized for better usability and user satisfaction**.

Lecture 8: Task Analysis in Interactive Systems Design

---

1. Overview of the Lecture

Lecture 8 of *Design of Interactive Systems (DIS)* focuses on **Task Analysis**, a crucial method in **Human-Computer Interaction (HCI)** for understanding user behavior and system interactions.

The lecture explains: ✔ **Goals, tasks, and actions** – How users interact with systems.
✔ **Hierarchical Task Analysis (HTA)** – A structured approach to breaking down tasks.
✔ **GOMS Model** – A cognitive model for predicting user interactions.
✔ **Structural Knowledge** – How users build mental models of systems.
✔ **Cognitive Work Analysis (CWA)** – An advanced framework for analyzing complex work environments.

Task analysis ensures **interactive systems are designed to support user workflows efficiently**.

---

2. Introduction to Task Analysis

> *"Task analysis is essential for understanding how people carry out their work with interactive systems."*

Why is Task Analysis Important?

- **Helps designers understand user needs and workflows**.
- **Identifies inefficiencies and bottlenecks**.
- **Improves usability by optimizing system interactions**.
- **Ensures the system supports users' mental models**.

✔ **Example:** Designing a **food delivery app**

- Users **search for restaurants, add items, and place orders**.
- Task analysis ensures the app **minimizes steps and provides clear feedback**.

---

3. Goals, Tasks, and Actions in Interactive Systems

> *"A task is a goal combined with an ordered set of actions."*

Understanding the Work System

✔ **Work System = Users + Technology + Environment**
✔ **Application Domain = The real-world problem the system addresses**

Example: A **hospital management system**

- **Work System:** Doctors, nurses, hospital database.
- **Application Domain:** Patient record-keeping, diagnosis management.

✔ **Task Analysis focuses on optimizing the Work System.**

A. Defining Goals

> *"A goal is the desired outcome a system or user wants to achieve."*

✔ **Example: Recording a TV show**

- **Current State:** The show is not recorded.
- **Goal:** Record the show for later viewing.
- **Possible Tasks:**
  1. Set a timer on the TV.
  2. Press the record button.
  3. Use a mobile app to schedule recording.

✔ **The system must help users reach their goals efficiently.**

B. Tasks vs. Actions

> *"Tasks are structured sets of activities, while actions are single steps within a task."*

✔ **Example: Booking a Cab in Uber**

Level	Example
Goal	Reach a destination.
Task	Book a cab using the Uber app.
Subtasks	Open app → Enter location → Choose car type → Confirm booking.
Actions	Tap "Book Now" → Wait for driver confirmation.

✔ **Tasks are broken down into subtasks, which eventually become individual actions.**

---

4. Hierarchical Task Analysis (HTA)

> *"HTA is a graphical method for representing task structures."*

A. What is HTA?

- **Breaks down tasks into subtasks and actions**.
- **Uses a structured diagram** to show task flow.
- **Helps identify inefficiencies in task execution**.

✔ **Example: Making a Call Using a Mobile Phone**

Task	Subtasks	Actions
Make a call	Find contact	Open contacts, search name
	Dial manually	Open keypad, enter number
	Press "Call"	Tap the call button

✔ **HTA improves user experience by identifying unnecessary steps.**

---

5. GOMS Model – A Cognitive Task Analysis Method

> *"GOMS predicts how users interact with a system and estimates task performance."*

A. Components of GOMS

Component	Description	Example: ATM Withdrawal
Goals	What users want to achieve.	Withdraw money.
Operators	Physical and cognitive actions.	Insert card, enter PIN, press buttons.
Methods	Steps taken to complete the task.	Choose withdrawal amount, confirm transaction.
Selection Rules	Decision-making strategies.	Use quick withdrawal vs. manual entry.

✔ **GOMS is used in UI optimization to reduce task time and complexity.**

---

6. Structural Knowledge and Mental Models

> *"Users form mental models of how a system works, influencing how they interact with it."*

A. Goal Space vs. Device Space (Payne, 2012)

✔ **Goal Space** – What users want to do.
✔ **Device Space** – How the system enables actions.

✔ **Example: Using a Drawing App**

- **Goal Space:** Increase brush size.
- **Device Space:** Locate brush settings menu.

If **users struggle to find the brush settings**, **there's a usability problem**.

---

7. Cognitive Work Analysis (CWA) – Advanced Task Analysis

> *"CWA is used for analyzing mission-critical environments like power plants and aviation."*

A. Key Principles of CWA

✔ **Analyzes real-time, high-risk work environments**.
✔ **Focuses on system adaptability and decision-making under pressure**.

✔ **Example: Air Traffic Control System**

- **Operators must process high volumes of information quickly**.
- **CWA ensures the system supports fast and accurate decision-making**.

✔ **CWA is essential for designing interfaces in high-risk environments**.

---

8. Task Analysis in System Design

✔ **Task analysis helps in:**

1. **User Research** – Identifying user needs and workflows.
2. **Interface Design** – Structuring UI elements based on tasks.
3. **Testing & Evaluation** – Identifying usability issues before deployment.

✔ **Example: Optimizing an Online Shopping App**

- **Task Analysis** identifies **bottlenecks in checkout flow**.
- **Improvements** – Reduce steps, provide autofill options.

✔ **Good task analysis = Better usability & efficiency.**

---

9. Class Activities

✔ **Activity 1: HTA for Purchasing a T-shirt from Myntra**

- **Break down the task into subtasks and actions**.

✔ **Activity 2: Write a GOMS Model for an ATM Transaction**

- **Define goals, operators, methods, and selection rules**.

---

10. Conclusion

✔ **Task analysis is crucial for designing intuitive interactive systems.**
✔ **HTA provides a structured breakdown of tasks and actions.**
✔ **GOMS predicts user performance and optimizes UI workflows.**

✔ **Structural knowledge ensures users form correct mental models.**
✔ **CWA is essential for high-risk, real-time environments.**

By applying task analysis techniques, **designers can create more efficient, user-friendly, and error-resistant interactive systems**.

Lecture 9: Visual Interface Design in Interactive Systems

---

1. Overview of the Lecture

Lecture 9 of *Design of Interactive Systems (DIS)* focuses on **Visual Interface Design**, an essential aspect of **Human-Computer Interaction (HCI)**. The visual interface is the **medium through which users interact with systems**, influencing usability, engagement, and accessibility.

The lecture discusses: ✔ **Types of interaction** – Command languages, graphical user interfaces (GUIs), direct manipulation.
✔ **Interface components** – Windows, icons, menus, pointers (WIMP).
✔ **Design principles** – Consistency, feedback, affordance, and error prevention.
✔ **Psychological factors** – Perception, memory, attention, and color usage.
✔ **Information design and visualization** – Presenting complex data effectively.

A **well-designed interface** enhances **efficiency, usability, and overall user experience**.

---

2. What is Visual Interface Design?

> *"The interface mediates the interaction between users and devices."*

A **user interface (UI)** consists of:

- **Physical Elements** – Buttons, touchscreens, keyboards.
- **Perceptual Elements** – Colors, icons, sounds, haptic feedback.
- **Conceptual Elements** – User expectations, mental models.

✔ **Example:** Designing a **mobile banking app**

- **Physical Interaction:** Users **tap the screen** to transfer money.
- **Perceptual Interaction:** Users **see a confirmation message**.
- **Conceptual Interaction:** Users **expect security and reliability**.

✔ **A good interface should be **intuitive, responsive, and visually appealing**.

---

3. Types of Interaction in User Interfaces

Users interact with systems in **three primary ways**:

A. Command Languages

> *"Command languages require users to input textual instructions."*

✔ **Examples:**

- **Unix/Linux CLI (Command Line Interface)**
- **SQL queries for databases**

✔ **Advantages:**

- Powerful and flexible for expert users.
- Quick execution with fewer steps.

✔ **Disadvantages:**

- Requires memorization.
- Not beginner-friendly.

✔ **Example:** Using a **Linux Terminal**

```
mkdir new_folder  # Creates a new directory
cd new_folder     # Navigates to the directory
```

B. Graphical User Interfaces (GUIs)

> *"GUIs use visual elements like icons, buttons, and windows."*

✔ **Examples:**

- Windows, macOS, Android, iOS
- Microsoft Office, Adobe Photoshop

✔ **Advantages:**

- Easy to learn (recognition-based, not recall-based).
- Interactive, intuitive, and visually guided.

✔ **Disadvantages:**

- Requires more resources (processing power, graphics).
- Sometimes slower than command-based systems.

✔ **Example: Microsoft Word UI**

- Click on the **Bold (B)** icon instead of typing a command.

C. Direct Manipulation Interfaces

> *"Users directly manipulate graphical objects instead of typing commands."*

✔ **Examples:**

- **Dragging files into folders** (file explorer).
- **Zooming in on a smartphone using pinch gestures**.
- **Interactive maps like Google Maps**.

✔ **Advantages:**

- Intuitive, visual, and engaging.
- Provides immediate feedback.

✔ **Disadvantages:**

- Can be complex for advanced operations.

✔ **Example: Using Google Maps**

- Drag the map to **navigate**, pinch to **zoom**, and tap locations for details.

---

4. The WIMP Model: Standard GUI Components

> *"Most modern interfaces are based on the WIMP (Windows, Icons, Menus, Pointers) model."*

A. Windows

✔ **Allow multitasking** by dividing the screen into **multiple areas**. ✔ Used in **desktop operating systems** and **web browsers**.

✔ **Example:** Opening multiple tabs in **Google Chrome**.

B. Icons

> *"Icons provide a visual representation of functions and files."*

✔ **Types of Icon Representation:**

Type	Example
Metaphor	Trash Bin icon for deleting files
Direct Mapping	Speaker icon for sound settings
Convention	Floppy disk icon for saving files

✔ **Good icons are:**
✅ **Recognizable** – Easily understood.
✅ **Consistent** – Follows established patterns.
✅ **Simple** – Minimal details, avoiding clutter.

✔ **Example:**

- A **shopping cart icon** universally represents **adding items for purchase**.

C. Menus

> *"Menus group commands into lists for easier selection."*

✔ **Types of Menus:**

Menu Type	Example
Hierarchical (Cascading)	Windows Start Menu
Pop-up Menus	Right-click menu in browsers
Contextual Menus	File-specific options in Finder

✔ **Example:**

- Right-clicking on **a file** in Windows **displays options like 'Open' and 'Rename'.**

D. Pointers

✔ **Pointers enable selection and navigation.**
✔ **Common types:**

- Mouse cursors
- Touch gestures (mobile)
- Stylus for tablets

✔ **Example:**

- Moving the **cursor over a hyperlink changes it into a hand icon**, indicating interactivity.

---

5. Design Principles for Effective Interfaces

> *"A well-designed interface follows usability principles to enhance user experience."*

✔ **Key Principles:**

Principle	Description	Example
Consistency	Uniform UI elements	Same color scheme in an app
Visibility	Important features should be prominent	Call-to-action buttons
Feedback	Immediate response to user actions	Loading animations
Affordance	Design should suggest function	Raised buttons for clickability
Error Recovery	Users should correct mistakes easily	Undo button in Word

✔ **Example:**

- **Gmail auto-saves drafts** to **prevent data loss**.

---

6. Psychological Factors in UI Design

> *"Users perceive, remember, and interact with interfaces based on cognitive principles."*

✔ **Key Cognitive Principles:**

Factor	Description	Example
Perception	Organizing visual elements	Grouping buttons in toolbars
Memory	Recognizing vs. recalling	Auto-suggestions in Google Search
Attention	Avoiding information overload	Clean UI layouts

✔ **Example:**

- **Google Search suggests previous queries**, reducing cognitive load.

---

7. Designing with Color

> *"Color enhances usability but must be used carefully."*

✔ **Best Practices:**

- **Limit color palette** to **5 ±2 colors**.
- **Use high contrast** for readability.
- **Follow cultural color conventions** (e.g., **red = danger**, **green = success**).

✔ **Example:**

- Traffic lights use **red, yellow, and green** universally.

---

8. Information Design and Visualization

> *"Complex data should be presented in an easy-to-understand format."*

✔ **Best Practices:**

- Use **charts, graphs, and maps** to summarize data.
- **Highlight key information** to avoid overwhelming users.
- Support **interactive visualization** (e.g., zoomable maps).

✔ **Example:**

- **Google Analytics uses charts** to display **website traffic trends**.

---

9. Error Handling & Alerts

✔ **Best Practices for Error Messages:** × **Avoid vague messages** (e.g., "Invalid input").
✅ **Provide clear solutions** (e.g., "Enter a valid email address").
✅ **Use non-threatening language** ("Oops! Something went wrong.").

✔ **Example:**

- **404 Error pages** should **suggest alternatives** instead of just displaying "Page Not Found."

---

10. Class Activity

✔ **Find examples of "Recall vs. Recognition" in apps you use.**

- Example: **Auto-fill forms vs. manually entering data.**

Conclusion

✔ **A well-designed interface ensures usability, accessibility, and efficiency.**
✔ **Visual UI components (WIMP) improve interaction.**
✔ **Cognitive principles shape how users perceive and use interfaces.**
✔ **Information should be designed for clarity and interactivity.**
✔ **Error handling should be user-friendly and informative.**

By applying these **best practices**, **interfaces can be designed to be intuitive, efficient, and engaging.**

Lecture 10: Multimodal Interface Design in Interactive Systems

---

1. Overview of the Lecture

Lecture 10 of *Design of Interactive Systems (DIS)* explores **Multimodal Interface Design**, which goes **beyond traditional graphical interfaces** by incorporating: ✔ **Speech, touch, gesture, and sound** as interaction methods.
✔ **Augmented Reality (AR) and Virtual Reality (VR)** for immersive experiences.
✔ **Tangible User Interfaces (TUIs)** that allow physical interaction with digital systems.
✔ **Wearable computing and mixed reality systems**.

Multimodal interfaces **enhance user experiences** by **making interactions more intuitive and natural**.

---

2. Understanding Multimodal Interfaces

> *"Technologies today go far beyond screen-based systems."*

What is a Multimodal Interface?

✔ **A system that combines multiple input and output methods**.
✔ **Enables interaction through voice, touch, gestures, and haptics**.
✔ **Improves accessibility, engagement, and efficiency**.

✔ **Example:**

- **Smart Assistants (Alexa, Siri, Google Assistant)**
  🎤 **Voice commands** for control.
  📱 **Touchscreen** for manual input.
  🔊 **Audio feedback** for responses.

✔ **Goal: Make interactions more natural and flexible**.

---

3. Mixed Reality (MR): The Bridge Between Digital & Physical Worlds

> *"Mixed Reality (MR) combines real and virtual environments for enhanced interaction."*

✔ **Coined by Milgram et al. (1994)**, MR includes:

1. **Augmented Reality (AR)** – Digital objects overlay **real-world views**.
2. **Augmented Virtuality (AV)** – Real-world data integrates into **virtual spaces**.

✔ **Example:**

- **Snapchat AR Filters:** Add digital effects to real-world faces.
- **Microsoft HoloLens:** Displays **holographic data** in real environments.

✔ **MR Categories:**

Type	Example	Technology Used
Immersive MR	Full VR experiences	Oculus Quest, HTC Vive
Non-Immersive MR	AR overlays on screens	Google Glass, Mobile AR

✔ **Key Challenges:**

- **Alignment of real & virtual objects (Registration).**
- **Accurate spatial tracking.**

---

4. Head-Mounted Displays (HMDs)

> *"HMDs immerse users in virtual or augmented environments."*

✔ **Two Types of HMDs:**

Type	Description	Examples
Video See-Through HMDs	Camera captures real-world, overlays digital content	HoloLens, Magic Leap
Optical See-Through HMDs	Transparent displays overlay digital data on real-world views	Google Glass

✔ **Popular HMDs:**

- **Oculus Quest** (VR gaming, 360° movies).

- **HTC Vive** (Virtual training & simulations).
- **Microsoft HoloLens 2** (Enterprise AR solutions).

✔ **Example:**

- **Surgeons using AR HMDs** for **overlaying digital scans** during operations.

---

5. Haptics & Touch in Multimodal Interfaces

> *"Haptics simulate the sense of touch in digital interactions."*

✔ **Types of Haptic Feedback:**

Haptic Type	Example
Vibration Feedback	Smartphone touch response
Force Feedback	Game controllers with resistance
Surface Texture Simulation	Virtual reality gloves

✔ **Example:**

- **PlayStation 5 DualSense Controller** – Provides **realistic haptic feedback** in gaming.

✔ **Challenges in Haptic Design:**

- Precision in **simulating textures**.
- **Latency** in real-time feedback.

---

6. Gesture-Based Interaction

> *"Gestural interaction enables users to control systems through movement."*

✔ **Common Gestural Interfaces:**

Type	Example
Touch Gestures	Pinch to zoom, swipe to navigate
Hand Tracking	Leap Motion, Kinect
Body Gestures	Wii, VR motion tracking

✔ **Example:**

- **Microsoft Kinect** tracks full-body movements for **gaming & fitness apps**.

✔ **Challenge:**

- **Accurate gesture recognition across different lighting conditions**.

---

7. The Role of Sound in Multimodal Interfaces

> *"Sound enhances interactions by reducing visual overload and improving accessibility."*

✔ **Why Use Sound?**

1. **Reduces visual strain** – Less need to read screens.
2. **Provides ambient cues** – Alerts, notifications.
3. **Enhances accessibility** – Screen readers for visually impaired users.

✔ **Example:**

- **Google Maps voice guidance** – Enables hands-free navigation.

✔ **Challenges in Sound Design:**

- **Avoiding overwhelming users with excessive audio**.
- **Ensuring clarity in noisy environments**.

---

8. Earcons & Auditory Icons

> *"Audio cues enhance usability by providing feedback and navigation aids."*

✔ **Earcons:**

- Short, **abstract** sounds conveying system status.
- **Example:** Windows startup sound.

✔ **Auditory Icons:**

- **Real-world sounds** representing actions.
- **Example:** Trash bin sound when deleting a file.

✔ **Study Findings:**

- Users **identified navigation sounds with 80% accuracy**, proving effectiveness.

---

9. Speech-Based Interfaces (SBI)

> *"SBIs enable users to interact with systems using natural language."*

✔ **Examples:**

- **Amazon Alexa** – Smart home voice control.
- **Google Assistant** – Voice-driven search & automation.

✔ **Key Components:**

Component	Function
Automatic Speech Recognition (ASR)	Converts voice to text

Component	Function
Text-To-Speech (TTS)	Converts text to voice
Natural Language Processing (NLP)	Interprets user intent

✔ **Challenges:**

- **Accents, background noise, speech ambiguity.**

✔ **Example:**

- **Siri vs. Alexa – Which AI understands commands better?**
  - **Alexa** excels at **smart home automation**.
  - **Siri** offers **deeper Apple ecosystem integration**.

---

10. Tangible User Interfaces (TUIs)

> *"TUIs integrate physical objects with digital systems."*

✔ **Example TUIs:**

- **Microsoft Surface** – Supports pen input & gesture recognition.
- **Reactable** – Music composition using tangible blocks.

✔ **Key Benefit:**

- **Bridges the gap between physical and digital interactions**.

✔ **Challenge:**

- **Ensuring accurate mapping of physical movements to digital actions**.

---

11. Information Design in Multimodal Systems

✔ **Best Practices for Multimodal Interfaces:**

Principle	Explanation
Redundancy	Use multiple channels (e.g., text + voice).
Minimize cognitive load	Avoid overwhelming users with too much information.
Adaptive systems	Customize UI based on user preferences.

✔ **Example:**

- **Tesla's voice + touchscreen interface** optimizes driver interactions.

---

12. Challenges in Multimodal Design

✔ **Key Challenges:**

1. **Context Awareness** – Recognizing user intent in different scenarios.
2. **Integration Complexity** – Synchronizing multiple interaction methods.
3. **Error Handling** – Providing smooth fallback options when one modality fails.

✔ **Example:**

- **Self-driving cars** combine **visual (cameras), auditory (alerts), and haptic feedback (steering vibrations).**

---

13. Class Activity

1. Suggest three different ways in which information could be displayed using sound.
2. Describe potential disadvantages of augmenting the interface with sound.

---

14. Conclusion

✔ **Multimodal interfaces enhance user experience through multiple interaction methods.**
✔ **Mixed Reality (MR) creates immersive digital-physical environments.**
✔ **Haptics, gestures, and speech-based interfaces improve accessibility and engagement.**
✔ **Sound-based cues (earcons & auditory icons) reduce visual load.**
✔ **Tangible User Interfaces (TUIs) blend physical and digital interactions.**

By integrating these elements, **interactive systems become more intuitive, efficient, and user-friendly.**

Lecture 11: Memory and Attention in Interactive Systems Design

---

1. Overview of the Lecture

Lecture 11 of *Design of Interactive Systems (DIS)* explores **Memory and Attention**, two fundamental aspects of **human cognition** that directly impact how users interact with interactive systems.

The lecture discusses: ✔ **Memory types and processes** – Short-term, long-term, and working memory.
✔ **Forgetting mechanisms** – Decay theory, interference, accessibility vs. availability.
✔ **Attention and cognitive load** – Selective and divided attention, mental workload.
✔ **Signal detection theory (SDT)** – Identifying important information amidst noise.
✔ **Human error and action slips** – Common mistakes and strategies to prevent them.

Understanding **memory and attention** helps designers create **user-friendly, error-resistant, and efficient systems**.

---

2. The Role of Memory in Interactive Systems

> *"Memory is not just a passive storage system; it actively processes and retrieves information."*

Types of Memory

✔ Memory stores information in different ways:

Memory Type	Function	Example
Working Memory	Temporary storage for active thinking.	Remembering a phone number while dialing.
Short-Term Memory	Holds limited information for a brief period.	Memorizing a shopping list.
Long-Term Memory	Stores knowledge for extended periods.	Remembering how to ride a bicycle.

✔ Example:

- When entering a **password**, **working memory** holds it temporarily.
- If the password is reused often, it moves to **long-term memory**.

✔ Challenge:

- Users often **forget passwords** due to **interference or lack of rehearsal**.

---

3. Working Memory and Cognitive Load

> *"Working memory processes active information but has strict capacity limits."*

✔ Working memory consists of:

Component	Function
Central Executive	Decision-making, attention control.
Phonological Loop	Stores auditory information (e.g., repeating a phone number).
Visuo-Spatial Sketchpad	Stores visual and spatial data (e.g., remembering maps).

✔ Example:

- When using **Google Maps**, the **Visuo-Spatial Sketchpad** helps in **remembering directions**.

✔ Limitations:

- Working memory **can only hold 5-9 chunks of information at a time** (**Miller's Law**).
- If overloaded, users **struggle to process information efficiently**.

✔ Design Implication:

- **Reduce clutter in interfaces** to prevent cognitive overload.

---

4. Long-Term Memory: Encoding and Retrieval

> *"Long-term memory stores vast amounts of information over time."*

✔ **Memory encoding methods:**

Type	Example
Semantic Memory	Remembering facts (e.g., Paris is the capital of France).
Episodic Memory	Remembering personal experiences (e.g., first day at college).
Procedural Memory	Skills-based knowledge (e.g., typing on a keyboard).

✔ **Example:**

- **Users remember common icons** (e.g., the "trash bin" for deleting files) through **semantic memory**.

✔ **Challenges:**

- Memory **fades over time** (Decay Theory).
- New information **interferes with old memory** (Interference Theory).

✔ **Design Tip:**

- **Use familiar metaphors and consistent UI patterns** to improve memory recall.

---

5. How and Why Do We Forget?

> *"Forgetting occurs due to decay, interference, or retrieval failure."*

✔ **Forgetting Theories:**

Theory	Description	Example
Decay Theory	Memory weakens over time without use.	Forgetting old phone numbers.
Interference Theory	New information overwrites old memories.	Learning a new language makes it harder to recall an old one.
Retrieval Failure	Information is stored but difficult to access.	Forgetting a word but recalling it later.

✔ **Example:**

- **Users forget passwords** due to lack of **retrieval cues**.

✔ **Solution:**

- **Provide hints or password managers** to aid memory.

---

6. Attention in Interactive Systems

> *"Attention determines how users process and respond to information."*

✔ **Types of Attention:**

Type	Example
Selective Attention	Focusing on a specific task while ignoring distractions (e.g., reading in a noisy café).
Divided Attention	Performing multiple tasks simultaneously (e.g., driving while talking).

✔ **Example:**

- **Using a smartphone while walking** divides attention, increasing accident risks.

✔ **Design Implication:**

- Reduce **cognitive distractions** by using **minimalist UI design**.

---

7. Mental Workload and Stress in UI Design

> *"Mental workload refers to the cognitive effort required to complete tasks."*

✔ **High workload = more errors & reduced efficiency.**

✔ **NASA Task Load Index (NASA-TLX)** measures:

- **Mental Demand**
- **Physical Demand**
- **Time Pressure**
- **Performance**
- **Effort**
- **Frustration Level**

✔ **Example:**

- Pilots use **Head-Up Displays (HUDs)** to reduce cognitive workload while flying.

✔ **Design Tip:**

- **Minimize UI complexity** to reduce mental workload.

---

8. Visual Search and Interface Design

> *"Users scan interfaces to locate information efficiently."*

✔ **Factors Affecting Visual Search:**

Factor	Example
Size & Brightness	Large, bright elements are easier to find.

Factor	Example
Positioning	Users scan from **left to right** in Western cultures.
Motion & Animation	Flashing alerts grab attention.

✔ **Example:**

- **Google Search highlights keywords in bold** to improve visual scanning.

✔ **Design Tip:**

- Use **contrast and whitespace** to guide user attention.

---

9. Signal Detection Theory (SDT)

> *"Signal Detection Theory explains how users distinguish important signals from noise."*

✔ **Example:**

- A **security guard at an airport** must detect **weapons on an X-ray screen** despite distractions.

✔ **Challenges:**

- **False Positives:** Seeing a threat when none exists.
- **False Negatives:** Missing an actual threat.

✔ **Design Tip:**

- Use **alerts and feedback systems** to highlight important signals.

---

10. Human Error and Action Slips

> *"Human errors occur due to cognitive overload or misinterpretation of tasks."*

✔ **Common Action Slips:**

Type	Example
Capture Error	Typing an old password instead of a new one.
Description Error	Pressing the wrong button on a remote control.
Loss of Activation	Opening an app but forgetting the purpose.

✔ **Example:**

- **Users often enter incorrect passwords** due to muscle memory.

✔ **Design Tip:**

- **Provide error prevention & easy recovery options** (e.g., Undo Button).

---

11. Designing to Reduce Errors

✔ **Error Prevention Strategies:**

Strategy	Example
Constraints	Graying out unavailable options.
Affordances	Buttons should visually indicate clickability.
Recovery	Providing "Undo" for accidental deletions.
Feedback	Displaying error messages with solutions.

✔ **Example:**

- **Google Docs autosaves progress**, reducing accidental data loss.

---

12. Class Activity

1. **Identify examples of selective and divided attention in daily technology use.**
2. **Suggest three UI improvements to reduce memory load in mobile apps.**

---

13. Conclusion

✔ **Memory and attention shape user interactions in digital systems.**
✔ **Understanding cognitive limits improves UI design.**
✔ **Reducing mental workload enhances usability.**
✔ **Error prevention & recovery mechanisms improve user experience.**

By designing with **memory, attention, and cognitive limits in mind**, interactive systems **become more intuitive, efficient, and user-friendly**.

Lecture 12: Affective Computing in Interactive Systems

---

1. Overview of the Lecture

Lecture 12 of *Design of Interactive Systems (DIS)* focuses on **Affective Computing**, the integration of **human emotions** into interactive system design. The lecture explores: ✔ **The role of emotions in interactive systems** – How emotions influence user experience.
✔ **Theories of emotions** – Psychological models explaining human affect.
✔ **Affective computing** – Computers recognizing, responding to, and generating emotions.
✔ **Emotion recognition technologies** – Sensors, machine learning, and behavioral analysis.
✔ **Expressing and responding to emotions** – How interactive systems simulate or evoke affect.

Affective computing enhances human-computer interaction (HCI) by making systems more engaging, responsive, and personalized.

---

2. What is Affect in Interactive System Design?

> *"Affect describes emotions, moods, and sentiments that influence human behavior and decision-making."*

✔ **Emotions impact cognition, perception, and social interactions.**
✔ **Affect is non-cognitive (not related to reasoning) and non-conative (not related to intent).**

Types of Emotions:

✔ **Basic Emotions** – Fear, anger, surprise (short-term, intense).
✔ **Long-Term Emotions** – Love, jealousy, anxiety (sustained affect).

✔ **Example:**

- **A stressed driver** pays more attention to hazards.
- **A happy customer** leaves positive product reviews.

✔ **Affective computing helps systems **adapt to users' emotional states**.

---

3. The Role of Affective Computing in Interactive Systems

> *"Affective computing enables computers to detect, interpret, and respond to human emotions."*

✔ **Three aspects of affective computing:**

1. **Recognizing human emotions** – Analyzing speech, facial expressions, or physiological signals.
2. **Synthesizing emotions** – AI-driven avatars expressing emotional states.
3. **Eliciting emotional responses** – Games or media influencing user emotions.

✔ **Example:**

- **A car detecting driver stress** and **activating relaxation mode** (dimmed lights, soft music).

✔ **Why It Matters:**

- **Emotion influences decision-making, learning, and engagement.**
- **Personalized experiences improve user satisfaction.**

---

4. Theories of Emotion in Psychology

> *"Psychological theories explain how emotions are formed and processed."*

✔ **Key Theories:**

Theory	Description	Example

Theory	Description	Example
James-Lange Theory	Emotions result from bodily responses.	Running from danger → Fear.
Cannon-Bard Theory	Emotions and bodily responses happen simultaneously.	Seeing a bear → Feeling fear + running.
Schachter-Singer Theory	Emotions arise from cognitive interpretation of bodily signals.	Heart racing → Interpreted as excitement or fear depending on the situation.

✔ **Example:**

- **Smartwatches track heart rate and detect stress, adjusting notifications accordingly.**

✔ **Design Implication:**

- Systems should **interpret physiological data within context** to avoid false alerts.

---

5. Detecting and Recognizing Emotions

> *"Emotional states have physiological, cognitive, and behavioral components."*

✔ **Physiological signals used for emotion detection:**

Signal	Measurement	Example Use
Facial expressions	Camera-based emotion analysis	Face ID detecting user mood.
Speech patterns	Tone, pitch, speed	Call centers detecting frustration.
Heart rate (ECG)	Wearable sensors	Fitness trackers monitoring stress.
Skin conductance (GSR)	Sweat gland activity	Lie detection tests.
Posture & gestures	Motion tracking	VR detecting body language.

✔ **Example:**

- **Amazon Alexa detects voice tone changes** to assess frustration levels.

✔ **Challenge:**

- **Similar physiological responses** occur for **different emotions** (e.g., fear vs. excitement).

✔ **Solution:**

- **AI combines multiple signals** (facial + vocal + physiological) for accuracy.

---

6. Emotion Recognition Technology: How Computers Understand Feelings

✔ **Key Capabilities Required:**

1. **Input:** Capturing emotional cues (voice, facial expressions, sensors).
2. **Pattern Recognition:** Identifying emotional signals.
3. **Reasoning:** Predicting emotions based on context.
4. **Learning:** Adapting to individual users over time.
5. **Output:** Generating an appropriate response (e.g., chatbot empathy).

✔ **Example:**

- **AI customer service detects frustration** and **offers personalized responses**.

✔ **Design Challenge:**

- Avoid **misinterpreting emotions** (e.g., mistaking excitement for anger).

---

7. Expressing Emotions in Interactive Systems

> *"Interactive systems can simulate or express emotions to improve engagement."*

✔ **Ways Computers Express Emotions:**

Method	Example
Animated Avatars	Virtual assistants (e.g., Siri, Google Assistant).
Haptic Feedback	Game controllers vibrating with in-game tension.
Sound & Tone Adjustments	AI changing voice tone based on context.
Visual Cues	Emojis, animations, color changes in UI.

✔ **Example:**

- **Robots like "Kismet" use facial expressions to convey emotions**.

✔ **Why It Matters:**

- **Users engage more with systems that feel responsive and "alive."**

---

8. Affective Wearables: Emotion-Sensing Devices

> *"Wearables monitor physiological signals to detect emotional states."*

✔ **Examples of Affective Wearables:**

Device	Function
Smartwatches (Apple Watch, Fitbit)	Stress & heart rate tracking.
Emotion-Sensing Jewelry	Detects mood via skin temperature.
Brainwave-Reading Headbands	Measures focus & relaxation (e.g., Muse Headband).

✔ **Example:**

- **Blood Volume Pressure (BVP) earrings** measure heart rate for **emotion analysis**.

✔ **Future Applications:**

- **Smart clothing that adapts to mood (e.g., color-changing fabrics).**

---

9. Emotional AI in Interactive Systems

> *"Emotional AI enhances user experience by personalizing interactions."*

✔ **Applications of Emotional AI:**

Domain	Application
Gaming	AI adjusts difficulty based on player frustration.
Healthcare	AI detects depression from speech patterns.
Marketing	Ads adapt to detected emotions.
Education	Personalized learning experiences based on student engagement.

✔ **Example:**

- **AI in gaming monitors frustration** and **adjusts game difficulty dynamically**.

✔ **Challenge:**

- **Ethical concerns about privacy and emotional data collection.**

---

10. Ethical Considerations in Affective Computing

> *"Emotion recognition raises privacy, bias, and ethical concerns."*

✔ **Key Issues:**

1. **Privacy Risks** – Emotion data is personal and sensitive.
2. **Bias in AI Models** – Systems may misinterpret emotions across different cultures.
3. **User Manipulation** – Companies could exploit emotions for marketing.

✔ **Example:**

- **Facebook's emotional AI** influences ad targeting based on user moods.

✔ **Design Solution:**

- **Transparent AI models with user control over emotional data collection.**

---

11. Class Activity

1. How could wearable emotion sensors enhance gaming experiences?
2. What ethical risks arise from AI detecting human emotions?

---

12. Conclusion

✔ **Emotions influence user interactions, decision-making, and engagement.**
✔ **Affective computing enables machines to recognize, simulate, and respond to emotions.**
✔ **Emotion recognition uses AI, sensors, and behavioral analysis for accuracy.**
✔ **Interactive systems can express emotions using avatars, sound, and visuals.**
✔ **Ethical concerns must be addressed to prevent misuse of affective technologies.**

By integrating **affective computing**, interactive systems **become more human-like, personalized, and intuitive**.

TILL MIDSEM : Lecture 13: Cognition and Action in Interactive Systems Design

---

1. Overview of the Lecture

Lecture 13 of *Design of Interactive Systems (DIS)* explores **Cognition and Action**, focusing on how humans think, perceive, and interact with technology. The lecture discusses: ✔ **Human Information Processing (HIP)** – How humans process and respond to digital systems.
✔ **Norman's Seven-Stage Model** – The mental process behind human actions.
✔ **Distributed Cognition** – How cognition is spread across people, tools, and systems.
✔ **Embodied and Enactive Interaction** – The role of body movement and environment in cognition.
✔ **Activity Theory** – A framework for understanding human activities in interactive system design.

Understanding **cognition and action** helps designers build **efficient, intuitive, and error-free interfaces**.

---

2. What is Cognition in Interactive System Design?

> *"Cognition includes all conscious and unconscious processes by which knowledge is accumulated."*

✔ **Cognition includes:**

- **Perception** (Seeing a stop sign and recognizing danger).
- **Attention** (Focusing on reading despite background noise).
- **Memory** (Recalling passwords for login).
- **Decision-Making** (Choosing a product online).
- **Problem-Solving** (Figuring out software errors).

✔ **Example:**

- When a user **navigates a website**, cognition helps them **perceive UI elements, interpret information, and take action**.

✔ **Design Implication:**

- **Cognitive overload reduces efficiency**, so interfaces should be **minimalist and structured**.

---

3. Human Information Processing (HIP)

> *"Human-Computer Interaction (HCI) relies on models of human cognition to optimize design."*

✔ **HIP Model consists of:**

Component	Function
Sensory Input	Receives information (vision, hearing, touch).
Information Processing	Analyzes and interprets data.
Motor Output	Executes actions (typing, clicking, swiping).

✔ **Example:**

- **Using an ATM:**
  1. **Sensory Input:** Seeing the ATM screen.
  2. **Processing:** Deciding which button to press.
  3. **Motor Output:** Pressing the "Withdraw" button.

✔ **HIP is useful for interface design but has limitations**: × Too **simplistic** – Humans are more than just "information processors."
× **Ignores social and emotional factors** in decision-making.

---

4. Norman's Seven-Stage Model of Action

> *"Human actions follow a structured mental process before execution."*

✔ **Seven Stages of Activity:**

Stage	Example: Checking Sports Results
1. Goal	"I want to check match scores."
2. Intention	"I need to open a sports app."
3. Action Planning	"I will unlock my phone and tap the app."
4. Execution	Opens the app, searches for the score.
5. Perception	Sees the displayed scores.
6. Interpretation	Understands if their team won.
7. Evaluation	Decides whether to celebrate or check highlights.

✔ **Challenges in Execution:**

- **The Gulf of Execution:** Difficulty in performing an action.
  - Example: **Struggling to find a sports app on a cluttered phone screen.**
- **The Gulf of Evaluation:** Difficulty in interpreting system feedback.
  - Example: **Not knowing whether the app is refreshing data or frozen.**

✔ **Design Tip:**

- **Reduce cognitive gaps** by making actions and feedback **clear, predictable, and intuitive.**

---

5. Why HIP Alone is Not Enough

> *"Traditional cognitive models (HIP) do not capture the complexity of human behavior."*

✔ **Limitations of HIP Models:** × **Too simplistic** – Human cognition is not linear.
× **Ignores social context** – People interact with technology socially.
× **Fails to explain real-world problem-solving** – Learning and adaptation play a role.

✔ **Alternative Cognitive Models:**

- **Distributed Cognition**
- **Embodied Cognition**
- **Activity Theory**

---

6. Distributed Cognition: Thinking Beyond the Brain

> *"Cognition is shared across people, tools, and environments."*

✔ **Example: The Moon Landing (1969)**

- Astronauts Armstrong and Aldrin **landed on the Moon**, assisted by **Mission Control in Houston**.
- **Cognitive processes were distributed**:
  - **Astronauts** focused on flying.
  - **Ground control** provided navigation guidance.

✔ **Other Examples:**

Scenario	Distributed Cognition
Driving in a new city	GPS provides directions, passengers assist.
Shopping in a supermarket	A list + shelf arrangements aid decision-making.
Collaborative work on spreadsheets	Employees + Excel formulas work together.

✔ **Design Implication:**

- Systems should **support collaboration and shared cognitive resources**.

---

7. Embodied Cognition: Thinking with the Body

> *"Cognition is influenced by physical actions and the environment."*

✔ **Key Concepts:**

- **Physical objects shape thought** (e.g., using an abacus to do math).

- **Action influences perception** (e.g., feeling the weight of an object before lifting).
- **People interact with technology through bodily movement.**

✔ **Example:**

- **Touchscreens allow direct manipulation**, improving engagement.

✔ **Design Tip:**

- **Use natural interactions like swiping, pinching, and dragging.**

---

8. Affordances in Interactive Design

> *"Affordances are properties of objects that suggest how they should be used."* – Don Norman

✔ **Examples of Affordances:**

Type	Example
Physical Affordance	A door handle invites pulling.
Cognitive Affordance	Play icon (▶) suggests clicking.
Perceived Affordance	Raised buttons look pressable.

✔ **Design Tip:**

- **Good UI elements should suggest their function clearly.**

✔ **Example:**

- **A shopping cart icon** naturally conveys the function of adding items.

---

9. Activity Theory: Understanding Human Actions

> *"All human activities involve an interaction between a subject, tools, and an object (goal)."*

✔ **Example: Learning to Drive**

Component	Example
Subject	A person learning to drive.
Tools	Steering wheel, pedals, GPS.
Object (Goal)	Successfully driving on a highway.

✔ **Activity Breakdown:**

1. **Actions** – Turning the key, shifting gears.
2. **Operations** – Subconscious habits like checking mirrors.

✔ **Over time, actions become automatic, reducing cognitive load.**

✔ **Design Tip:**

- Design systems that **guide beginners** but also offer **shortcuts for experts**.

---

10. Class Activities

1. **Identify a device you find difficult to use.**
   - What **gulf of execution** or **gulf of evaluation** exists?
2. **Find an example of misleading affordances in daily life.**
   - How does the design create confusion?

---

11. Conclusion

✔ **Cognition influences how users process, interpret, and interact with technology.**
✔ **Norman's Seven-Stage Model explains human decision-making and action.**
✔ **Distributed cognition shows how thinking is shared across people and tools.**
✔ **Embodied cognition highlights the importance of physical interaction.**
✔ **Activity theory explains how actions become automated over time.**
✔ **Affordances should be designed to guide users intuitively.**

By applying **cognitive models to interactive systems**, designers can **optimize usability, reduce errors, and enhance engagement.**

Lecture 14: Foundations of Designing Interactive Systems – Social Interaction

---

1. 📜 Overview of the Lecture

This lecture—taught by **Dr. Kalpana Shankhwar (IIIT-Delhi)**—moves beyond the ergonomics of individual interfaces and asks a tougher question: **"How does the *social* nature of human beings reshape the very core of interactive-system design?"** The slide-deck frames its discussion around Chapter 24 of *Part IV* (Memory & Attention → Perception & Navigation) and zooms into four pillars: **human communication, group participation, presence, and culture/identity** .

---

2. Human Communication – The Bedrock of Social UX

2.1 Semiotics & Sign Theory

- A *sign* couples a **signifier** (form) with a **signified** (concept)—e.g., the 🔔 icon in a UI (form) evokes "notifications" (concept).
- Designers manipulate *channels* (visual, auditory, haptic) to ferry signs from **transmitter** to **receiver** .

> **Quote:** "Social interaction begins with the ability to communicate."

2.2 Linguistic vs Non-Verbal Layers

Layer	Design Take-away	Key Examples
Linguistic (words, syntax)	Provide clear affordances; support multilingual text-entry.	
Non-Verbal (NVC) – movement, eye-gaze, prosody	Build cameras/mics capable of capturing *paralinguistic* cues for richer remote collaboration .	Emoji reactions 👍, eye-tracking cursors in Figma LiveShare

2.3 Prosody & Speech Interfaces

Prosody (pitch, rhythm, stress) conveys emotion often absent in text. Smart speakers that ignore prosody risk misinterpreting sarcasm or urgency .

2.4 Facial Expressions, Gesture, Body Language

- **Facial expressions** are high-bandwidth channels; entire cortical regions decode them .
- **Gestures** supplement deictic references—pointing at a whiteboard clarifies "*this* variable."
- **Body posture** signals attitude (lean-in = engagement). UI avatars that mimic torso lean in VR can increase trust.
- **Design Tip:** Support **pose mirroring** in telepresence robots to retain these cues.

2.5 Proxemics – Designing with Space

Edward Hall's four zones (intimate 15-50 cm → public 3 m+) dictate comfort levels .

> **Example:** VR multiplayer apps should avoid spawning strangers inside a user's "intimate bubble" to prevent discomfort.

2.6 Common Ground

Olson & Olson's framework (co-presence, co-temporality, visibility, etc.) reminds us to offer **shared reference points**—e.g., a collaborative cursor or live pointer in Google Docs .

---

3. Participating in Groups – Dynamics & Pitfalls 🧑‍🤝‍🧑

3.1 Group Lifecycle

Most groups travel predictable phases (forming → storming → norming → performing). UIs that visualize *team progress* can nudge transitions more smoothly .

3.2 Social Norms & Productivity

The classic **Bank-Wiring Room** study showed productivity spiked because workers felt *observed yet autonomous*, not because of lighting tweaks . Lesson: **perceived autonomy** and **peer approval** outweigh many ergonomic tweaks.

3.3 Compliance – The Stanford Prison Reminder

Role assignment can override morals; guards adopted cruelty within days . Design ethics demand **safeguards against emergent abuse** (e.g., moderation dashboards, rate-limit levers).

3.4 Groupthink & Risk Shift

Teams may accept extreme risks (Apollo XI's ~50 % survival estimate) under cohesion pressure . A good collaboration tool should encourage **"red-team" workflows** to surface dissent.

3.5 Conformity – Asch's Line Experiment

32 % of individuals nodded to an obviously wrong majority answer . UI implication: reveal *independent* voting results only after users commit, preventing herding in polls.

3.6 Productivity Traps: Social Loafing & Production Blocking

Brainstorming in one Zoom channel yields fewer ideas than parallel solo ideation; asynchronous whiteboards (Miro stickies) combat **production blocking** .

3.7 Technology to the Rescue?

Group Decision Support Systems (GDSS)—e-mail conferencing, anonymous voting—boost participation breadth but prolong decision time . Designers must balance **diversity** with **speed** via smart facilitation features.

---

4. Presence & Telepresence – Feeling "There" 🌘

Facet	Definition	Design Implication
Presence	Subjective sense of "being there"	High-fidelity visuals help, but **narrative engagement** matters just as much .
Co-Presence	Being *with* others	Spatial audio + avatars that echo gaze build "shared space."
Breaks in Presence	Cyber-sickness, network lag	Provide latency-tolerant interaction loops (e.g., predictive motion smoothing) .

> Chandrayaan-3's remote operators need *deep* telepresence to land on the Moon; milliseconds matter .

---

5. Culture & Identity – Designing for the Global Mosaic 🌐

5.1 Hofstede's Five Dimensions

1. **Power Distance** – hierarchy vs egalitarian UI ("admin-only" functions).
2. **Individualism ↔ Collectivism** – privacy controls vs shared dashboards.
3. **Masculine ↔ Feminine** – competitive badges vs community-care signals.
4. **Uncertainty Avoidance** – progress bars, confirmation dialogs.
5. **Long- vs Short-Term Orientation** – subscription models vs one-off purchases .

5.2 Localization Beyond Translation

Icons, metaphors, even color palettes carry cultural weight (e.g., red = luck in China, danger in the West) .

5.3 Digital Identity

The *"profile self"* (Instagram) intersects with group identities (Reddit communities) creating **multi-faceted personas** . Systems must allow **contextual identity presentation**—professional on LinkedIn, playful on Discord.

---

6. Design Imperatives – Turning Theory into Pixels

Challenge	Concrete Guideline
Lost NVC in video-calls	Integrate *gaze-correction* & *hand-gesture recognition* to transmit subtle cues.
Groupthink	Auto-assign a "devil's advocate" role in sprint-planning apps.
Social loafing	Display *individual* contribution metrics in collaborative docs, but reward *team* milestones.
Culture clash	Offer **"cultural presets"** (date formats, reading direction, color schemes) during onboarding.
Presence breaks	Provide manual **"re-center"** controls & gradual scene transitions to reduce cybersickness.

---

7. 📌 Key Takeaways

- **Communication is multimodal; design for *all* channels, not just text.**
- **Group dynamics can sabotage or supercharge productivity; tech must amplify healthy norms while damping bias.**
- **Presence is psychological; fidelity is necessary but *not sufficient* for "being there."**
- **Culture and identity demand flexible, localized experiences or risk alienation.**

> Ultimately, an interactive system is **never** just an interface—it is a **social organism** living inside the complex ecosystem of human norms, emotions, and cultures.

(End of in-depth analysis)

Paper Deep-Dive: "An Interactive Extended-Reality Tutorial System for Manual Metal-Arc Welding"

1️⃣ Scope & Rationale

Manual-metal-arc welding (MMAW) is **hazardous, parameter-rich, and expensive** to teach on the shop floor. The authors ask a bold question:

> *Can extended-reality (XR) replicate both the **cognitive** and the **psychomotor** sides of welding training—while slashing risk and material waste?*

They answer by building a two-part XR tutor—**VR for theory**, **MR for hands-on practice**—and benchmarking it against a traditional classroom.

2️⃣ Related-Work Trajectory

Domain	Past Insights	Gap Identified
XR in Education	3-D visualisation boosts spatial understanding but rarely covers *dangerous* skills (e.g., chemistry AR, dentistry VR)	Lacks empirical welding data.
XR in Manufacturing	MR improves assembly & HRI; commercial XR weld sims exist (VRTEX, Soldamatic)	Few give **real-time bead geometry feedback** or robust pedagogy for novices.
Welding Training	Simulators assess experts more than they **teach** rookies; cost & tracking issues persist	Need affordable tutor with parameter visualisation + autonomous guidance.

3️⃣ System Architecture Overview

Hardware

- **HTC Vive Pro** HMD + lighthouses
- **Vive Tracker** strapped to a real electrode holder
- **VL6180X distance sensor** on tip (measures arc length)
- **3-D-printed joints** (butt, tee, corner, lap)

Software

- **Unity 3D + C#**
- **VIU Toolkit** for controller/Tracker I/O
- **SRWorks SDK** for passthrough MR fusion

4️⃣ VR Module 🎓 — Conceptual Grounding

Learners enter a **fully-virtual classroom** where every critical MMAW concept is an interactive 3-D widget:

Welding science playlist

1. Joint & weld types
2. Arc physics (arc length ↔ voltage)
3. Process parameters (current, speed, electrode Ø, orientation)

4. Bead-geometry cause–effect loops

Pedagogical spice: hover-over tooltips, narrated audio, and teleport hotspots keep cognitive load low while ensuring engagement.

5️⃣ MR Module 🔧 — Kinesthetic Skill-Building

Learners walk to a physical bench; passthrough video overlays:

Feature	Implementation Nuance	Learning Payoff
Real Electrode + Tracker	6-DoF pose + sensor-based gap detection	Natural muscle memory for arc striking.
Dynamic Bead Simulator	Heat-input model: (Q = \frac{IV}{v}) governs bead width/penetration in real time	See parameter tweaks instantly —*no slag, no burns*.
Visual Aids	Ghost path (speed), cone HUD (angle), colour-coded arc gap	Immediate error-correction feedback.

6️⃣ User Study Design

Participants — 30 undergrads (non-ME) ⇒ XR group = 15, Control = 15
Protocol

1. Tutorial (XR vs slides)
2. **20-min written test** (max = 32 pts)
3. **NASA-TLX** workload survey
4. XR cohort attends classroom too → **SUS** + subjective Likert survey

---

7️⃣ Results & Interpretation

Metric	XR	Control	Insight
Written-test mean	**76 %**	44 %	XR triples retention on joint/parameter questions ($p < 0.001$)
NASA-TLX	↓ Mental & Temporal load; ↑ Physical, Effort, Frustration	-	Standing & novel UI raise physical strain, but cognitive load lower—mirrors authentic shop fatigue.
Performance sub-scale	Highest in XR	-	Real-time feedback accelerates mastery.
SUS score	72 (Grade B)	-	Good usability; learners want richer onboarding & Q&A.
Subjective	4.4 / 5 for “XR > classroom on interest & parameter clarity”	-	Immersive cause-effect loops trump slides.

8️⃣ Design Insights & Transferable Lessons

1. **Dual-stage XR (theory → practice)** beats mono-modal sims—cognitive scaffolding before motor skill rehearsal.
2. **Sensor-fused bead visualisation** turns invisible heat equations into visible learning moments.
3. **Effort ≠ Bad**—physical demand can *simulate real shop fatigue*, preparing learners realistically; just manage frustration via ergonomic hand-controllers & clearer onboarding.
4. **Cost & Safety ROI**: No consumables, zero UV exposure, reusable 3-D-printed joints—ideal for polytechnic budgets.

9️⃣ Limitations & Future Trails

- Tracking drift at electrode tip occasionally desynchronised bead rendering.
- No AI tutor for natural language Q&A (participants asked for it).
- Study used novices; expert skill-transfer longevity untested.

Next Steps – Integrate **inside-out HMD tracking**, embed **LLM-powered voice assistant** for on-the-fly coaching, and extend to **GMAW/TIG** parameter spaces.

🔑 Key Takeaways

- XR can deliver *both* declarative and procedural welding competence, **tripling knowledge retention** while minimising danger.
- Real-time, physics-based visual feedback on bead geometry is the killer feature.
- Usability is solid but **onboarding & conversational guidance** will unlock broader adoption.

(All quotations & factual details sourced from the uploaded PDF.)

Lecture 15: Perception & Navigation — Foundations for Designing Interactive Systems

---

1. 🎯 Lecture Overview

This session—taught by **Dr. Kalpana Shankhwar**—pairs two human super-powers that every interactive-system designer must respect: **perception** (how we *know* an environment) and **navigation** (how we *move* through it) . The material spans:

- Theories of visual perception & depth cues
- Gestalt laws & colour physiology
- Non-visual channels (audio / haptics)
- Way-finding principles for both physical and information spaces

---

2. 👁 Visual Perception

2.1 Constancies & Illusions

- **Colour constancy:** a red car remains "red" under daylight *and* sodium street-lamps—our brain discounts lighting .
- **Shape constancy:** a tilted coin is still recognised as circular.
- **Müller-Lyer & Necker cube** illusions show that perception is an *inference engine*, not a light sensor .

2.2 Gibson's Direct Perception

Pilots perceive **optic flow**—textures stream past a fixed cockpit point, delivering speed, altitude, and attitude data instantly . In UI terms, dynamic background parallax can cue motion without words.

---

3. 🏞 Depth Perception—From VR Caves to Flat Screens

Primary (Binocular)	Mechanism	Design Relevance
Retinal disparity + Stereopsis	7 cm eye separation → fused 3-D	Critical for immersive VR HMDs.
Accommodation & Convergence	Lens & ocular muscles report distance	HoloLens depth-aware UI.

Secondary (Monocular)	Practical Use-case
Light & shade	Drop-shadows on buttons.
Linear perspective	Infinite-scroll road in racing games.
Height in horizontal plane	Chess UI: black pieces "further" back .
Motion parallax	Foreground tweets scroll faster than background on mobile.
Overlap & relative size	Modal dialog overlaps dimmed page.
Texture gradient	Google Maps zoom transitions.

---

4. 🌀 Gestalt Laws—Why Layouts "Feel Right"

1. **Proximity** – cluster related toolbar icons.
2. **Continuity** – connect form steps with a progress bar arc.
3. **Part–whole (Emergence)** – logo reveals itself only after animation completes.
4. **Similarity** – colour-code categories.
5. **Closure** – dashed circle invites the mind to finish the shape .

---

5. 🎨 Colour Physiology & UX

- **Rods (≈120 M)** rule low-light, grayscale vision.
- **Cones (≈6–7 M)** concentrate in the **fovea**: 64 % red, 32 % green, 2 % blue—hence blue-on-black text is hard to read .

- Design takeaway: place crucial, high-resolution UI elements within the user's foveal hotspot—keep periphery low complexity.

---

6. 🔊 + ✋ Non-Visual Modalities

Modality	Key Parameters	UX Example
Auditory	Loudness (dB), frequency (Hz)	60 dB chat tone vs 110 dB alarm; pitch-scaled notifications for hierarchy .
Haptic	Tactile (skin) + kinaesthetic (muscles)	Smartphone "edge squeeze" gesture; VR controller recoil.

---

7. 🗺 Navigation: Object → Exploration → Way-Finding

- **Object identification:** recognise UI widgets.
- **Exploration:** learn page hierarchy / open-world map.
- **Way-finding:** reach a known goal quickly (e.g., checkout).

7.1 Design Principles for Navigable Spaces

1. **Distinctiveness over uniformity** – avoid identical corridors; colour-code web sections.
2. **Responsive environments:** multiple routes, clear landmarks, legible paths .
3. **Signage trio:**
    - *Informational* ("Profile completed 80 %")
    - *Directional* ("← Billing")
    - *Warning/Reassurance* ("Saved ✓")

---

8. Quick Heuristics for Designers

Goal	Ask Yourself…	Micro-Tweak
Reduce visual overload	"Can users group these as one chunk via proximity?"	Tighten spacing or add divider.
Convey depth on 2-D screen	"Which monocular cue fits?"	Add soft-shadow & linear-perspective grid.
Support ageing users	"Is pitch-based feedback still audible >12 kHz?"	Layer tactile pulse.
Way-finding in dashboards	"Is there a memorable landmark?"	Pin logo or avatar echo on every page.

🔑 Key Takeaways

- Perception supplies *meaning*, not raw pixels—leverage constancies & depth cues to craft intuitive 3-D illusions on flat displays.

- Gestalt laws are "design gravity"—ignore them and your UI feels chaotic.
- Navigation blends cognitive maps with emotional comfort; clear landmarks + responsive routes foster confidence.
- Multisensory channels (audio, haptics) compensate for visual limits—especially for accessibility and mobile contexts.

Master these perceptual building blocks, and your interactive systems will *feel* effortless, guiding users' eyes, ears, and bodies exactly where they need to go. 🧭

Paper Deep-Dive: "Learn Chemistry with Augmented Reality" (Macariu et al., 2020)

1️⃣ Why This Matters

Chemistry is *notoriously* abstract—molecules are invisible, reactions feel intangible. The authors ask:

> **"How many times did you have trouble understanding a new chemistry concept at school?"**

Their answer: **ARChemistry Learning**, a mobile AR ecosystem that turns textbooks, flash-cards, and quizzes into manipulable 3-D molecules—bridging *seeing* and *doing*.

2️⃣ Landscape Scan — Where Does This Fit?

Prior AR-in-education studies show increased engagement from anatomy to geography, yet **chemistry** implementations were piecemeal (e.g., Elements 4D cubes) and lacked integrated testing or teacher authoring. The paper positions AR as a *complement* (not replacement) to existing pedagogy .

3️⃣ System Architecture at a Glance

Layer	Tech	Role in Learning Flow
Marker Tracking	**Vuforia** image targets	Cards & textbook words become anchors for 3-D molecules.
3-D Engine	**Unity**	Renders ball-and-stick models; handles physics-based "attraction" when correct elements approach.
Mobile Platform	Android / iOS (smartphone or tablet)	Always-available pocket lab.
Singleton Substance DB	C# pattern	Allows real-time addition of *new* compounds by teachers/students—persisted across sessions.

Four user-facing modules :

1. **Learn with Manual** – point camera at textbook term → molecule + live Wikipedia snippet.

2. **Learn with Cards** – assemble Na + Cl cards → "snap" into NaCl with green outline.
3. **Test Your Knowledge** – timed challenges; score tallied (+ green / - red edges).
4. **Add a Substance** – teacher mode to create new markers & metadata.

4️⃣ Instructional Design Nuggets

Pedagogic Lever	Implementation Example
Dual Coding (text + visual)	Name + formula + coloured atoms overlay.
Immediate Feedback	Molecules attract only when stoichiometry correct; wrong combo glows **red**.
Gamification	Real-time scoring & next-level buttons in "Test" mode.
Teacher Co-Creation	Admin can inject curriculum-specific compounds on the fly.

5️⃣ Evaluation Methodology

Participants

- *Professors*: 70 (3-day trial)
- *Students*: 200 (1-day trial; ages mid- to high-school)

Tools

- **QUIS** (1–9 scale) for usability & "look-and-feel"
- Observation & task-completion logging

Key Findings

Metric	Professors	Students	Insight
"Overall look & feel"	**9.5/9**	**9.7/9**	AR visuals highly engaging.
Most loved feature	Bring Cards/Test (~9.1)	Same, plus spontaneous competition	Active composition > passive viewing.
Pain points	Lighting & camera angle needed for marker recognition; minor stability issues under *rapid* tapping	Similar; students sometimes "broke" flow finding shortcuts	Robustness & guidance layers required.

Qualitative gems:

- AR **reduced test anxiety**; students felt "like playing a game."
- Teachers valued visualising 3-D bonds that chalkboards can't convey.

6️⃣ Comparative Value vs Existing Apps

Solution	Limitation	ARChemistry Advantage
Elements 4D cubes	Fixed 36 elements, no quizzes	Dynamic compound creation + assessment.
Generic AR flashcards	Display-only	Wikipedia integration, scoring, teacher CRUD.

7️⃣ Limitations & Authors' Future Work

- Recognition needs good lighting; stability dips on rapid interactions.
- Desire for **voice-based Q&A** (planned Alexa integration) to allow hands-free responses .
- No long-term retention study yet.

8️⃣ Design Takeaways for Your Own AR Ed-Tech

1. **Blend passive & active modes**—view > build > test cycle strengthens memory.
2. **Provide teacher authoring** so content keeps pace with syllabus.
3. **Immediate multimodal feedback** (colour, motion, sound) cements correct answers.
4. **Prepare robust tracking**—include guidance UI for optimal marker distance & light.

🔑 Key Insights

- ARChemistry Learning shows **AR can transform abstract chemistry into a kinesthetic puzzle**, boosting enthusiasm for both learners and educators.
- Simplicity (paper cards + phones) makes large-scale classroom deployment feasible—even in resource-constrained settings.
- The model—visual cue ➜ interaction ➜ instant feedback ➜ assessment—can transfer to *any* STEM domain.

(All quotations & data points derive from the uploaded PDF.)

Lecture 16: Designing Websites

(Course: Design of Interactive Systems – Part III: Contexts for Designing Interactive Systems)

---

1. 📜 Setting the Stage – Why "Designing Websites" Matters

> "Usability and experience are crucial aspects in website design."

Designing a website is *not* an isolated art project—it is a socio-technical negotiation among **business strategy, user needs, information architecture, interface aesthetics, and code reality**. When a site delights, guides, and **never leaves the visitor wondering *"where am I?"***, it converts attention into action—whether that action is learning, purchasing, donating, or simply *coming back again*.

Key idea: *Every pixel is political.* Internal org politics, marketing agendas, and technical constraints surface in the final UI. If you ignore them, they will silently design the site for you.

---

2. Mapping the Terrain—DIS Context Chapters

Lecture 16 anchors a six-chapter arc (Ch. 14-19) exploring contexts from **websites** through **wearables**. Websites are the *gateway medium*—they influence how later contexts (mobile, ubiquitous, wearable) re-use patterns such as responsive grids, faceted search, and breadcrumb trails.

---

3. 🌐 Four Core Web Genres & Their DNA

Genre	Typical Goal	Content Pattern	Scrolling Habit	UX Pitfall
News	Time-sensitive updates	Dense article feeds, breaking bars	*Long* infinite scroll	Info overload
Shopping	Transaction & discovery	Product cards, filters, cart	*Short* paginated	Checkout friction
Information	Reference / learn	Hierarchical docs, FAQs	Mixed	Search burying facts
Entertainment	Engagement & return visits	Rich media, playlists	Variable	"Where did the video go?"

Different *sub-genres* (e.g., broadcast-TV vs. magazine sites) mutate these DNA strands—so cloning layouts blindly between genres invites usability debt.

---

4. 🎯 Lecture Aims—Three North Stars

1. **Process know-how:** stages from *strategy* → *evaluation*.
2. **Information Architecture (IA):** classification, labeling, mapping.
3. **Navigation Design:** bars, breadcrumbs, search harmony.

---

5. Pre-Design Reality Check (PACT + Strategy)

Before color palettes, ask: **Who is this for? How does it serve the org's digital strategy?** Disagreements and *"HiPPO"* (Highest Paid Person's Opinion) wars derail clarity. Many enterprise sites sprawl because marketing commandeers the homepage to chase every campaign. Remedy = **PACT analysis (People, Activities, Contexts, Technologies)** + **personas** that hold stakeholders accountable to *real* user stories.

---

6. ✍ The Hidden Skill—Writing for the Web

A web designer is *also* an **information author**. University sites often bloat trying to please *prospective students, current students, faculty, admin, external partners, alumni…* Result: link jungles. Personas + **content audits** prune dead pages and craft task-focused vocabularies.

> "Trying to accommodate all these different user groups results in an unruly and rambling site."

---

7. 🚀 The User-Centric Dev Loop

1. **Understanding** – field research, analytics.
2. **Envisionment** – scenarios, storyboards.
3. **Design** – low-fidelity → high-fidelity.
4. **Evaluation** – heuristic review, usability tests.

Scenarios of use flow into prototypes; **global top navigation + side nav** create a mental map; consistency is the cognitive shorthand the brain loves.

---

8. 🔗 Link Visibility & Entry Points

Links must *look* clickable—blue, underlined, or styled consistently. Plan for visitors who land on *any* page via Google; a prominent ***Home*** anchor rescues them. Jakob Nielsen's research: ≈ 50 % users are **search-dominant**, 20 % **link-dominant**, rest hybrid. Design both excellent **search** *and* **scannable menus**.

---

9. Jesse James Garrett's Five-Plane Model (2003)

Plane	Core Question	Deliverables	Gotcha
Strategy	*Why* are we doing this?	Vision doc, success metrics	Business ≠ user goals
Scope	*What* are we building?	Functional specs, content reqs	Feature creep
Structure	*How* is info organized?	Site map, IA diagrams	Confused mental model
Skeleton	*Interface framework?*	Wireframes, nav schema	Inconsistent affordances
Surface	*Visual polish?*	Final mock-ups	"Link should look like a link" → affordance clarity

Garrett's planes prevent teams leaping to *surface* (aesthetics) before nailing strategy.

---

10. 🗺 Site Maps & Wireframes

Garrett's "simple graphical language" uses icons for **pages**, **documents**, **stacks**, **diamonds** (decisions), **cross-bars** (forbidden). Wireframes stitch **information design, navigation, interface** into a skeleton blueprint—crucial for early stakeholder debates *before* engineers hard-code misaligned flows.

---

11. 🏛 Information Architecture (IA)—The Skeleton's Backbone

IA asks: **How do we classify, label, relate, and present content so that *any* user can retrieve *any* item with minimal cognitive load?** Three intertwined concerns:

1. **Organize** – build a *taxonomy* or *faceted* structure.
2. **Label** – choose user-centric, unambiguous terms.
3. **Describe & Expose** – metadata, previews, contextual links.

---

12. Implementation Tech Snapshot

Even in 2025, baseline page scaffolding remains **HTML + CSS + JavaScript**. *Dynamic HTML* (DOM manipulation) unlocked GUI-like drag-and-drop; legacy **Flash** once enabled miniature apps—now replaced by Canvas/WebGL. Knowing tech limits steers design realism.

---

13. 🗂 Classification Schemes—Choosing the Right Lens

A. Traditional Trio (Morville & Rosenfeld, 2006)

- Alphabetical
- Chronological
- Geographical

B. Shedroff's Seven (2001) adds *continuums, numbers, categories, randomness*.

Alphabetical quirks: Fail when formal vs. informal names differ (e.g., *"IBM"* vs. *"International Business Machines"*). Embed aliases and redirect links to avoid 404 frustration.

---

14. Ontology, Taxonomy & Epistemology 🤔

> "Ontology… concerns how we determine if things exist or not. Taxonomy… method of classification. Epistemology… how we come to know things."

Put simply:

- **Ontology:** *What's in the universe of our site?*
- **Taxonomy:** *How do we bucket those things?*
- **Epistemology:** *How will users recognize/learn those buckets?*

IA success = alignment of all three.

---

15. 📐 Organizational Structures

Structure	Best For	Caution
Hierarchy (Tree)	Broad topics drilled down – e.g., *Genre → Sub-genre → Album*.	Too many levels = "pogo-sticking".
Network (Poly-hierarchy)	Multi-category items (blog tags).	Users may lose sense of *place*.
Sequence (Wizard)	Step-by-step tasks (checkout).	Needs breadcrumbs & progress bar.

Rule of thumb: **6–8 links per branch** strikes balance between overview and choice paralysis.

---

16. Faceted Classification—Beyond One Dimensional Trees

A product (dimension) owns facets like **price**, **brand**, **rating**. Values filter dynamically. Think *Amazon's sidebar filters* or *recipe sites* (cuisine, ingredient, course). Facets empower *AND* searches without forcing users to navigate deep hierarchies.

---

17. 📑 Metadata & Controlled Vocabularies

Type	Purpose	Example
Intrinsic	Technical facts	*fileSize=2 MB; type=JPEG*
Administrative	Governance	*author=Kalpana; revised=2025-04-01*
Descriptive	Discovery facets	*genre=Jazz; mood=Upbeat*

Taxonomies like **Dewey Decimal** reveal synonym/homonym mines. A robust **thesaurus/ontology** plus **redirect pages** resolves "pop vs. soda" naming clashes.

---

18. 🧭 Navigation Design—Three Pillars

1. **Labelling** – crystal names; icons only where culturally obvious.
2. **Navigation Support** – global nav bars on *every* page, local sidebars, breadcrumbs (*"you are here"*).
3. **Search** – clear scope indicator, advanced filters for power users.

---

19. Brinck et al.'s Seven User Navigation Behaviours

Behaviour	Design Response
Omniscience	Offer fastest 1-2 click paths.
Optimal Rationality	Provide strong *information scent*—link text previews target.
Satisficing	Surface key actions above the fold.
Mental Maps	Keep topology simple; reuse patterns.
Rote Memorization	Stable URLs & landmarks aid repeat visits.
Information Foraging	Related-links, recommended articles.
Information Costs	Minimise clicks *and* cognitive guessing.

---

20. 🔖 Labelling Consistency & "Village Idiot" Test

Switching label synonyms (*"products"* ↔ *"items"*) confuses. Run a **First-Click Test**: if 80 %+ users pick the intended link in < 5 s, label passes.

---

21. 🔍 Search UX—Scope & Syntax

Two pain points:

1. *What exactly is being searched?* Indicate with placeholders (*"Search articles only…"*).
2. *How to express complex queries?* Provide guided filters (date range, file-type toggles) to avoid Boolean-syntax brain-strain.

---

22. 🎓 Class Activity Inspiration

Compare **IIT Delhi** vs. **IIIT Delhi** sites on Garrett's **surface** plane. Examine:

- Visual hierarchy of global nav bars
- Breadcrumb implementation
- Faceted vs. hierarchical menus
- Mobile responsiveness

Bring screenshots, annotate successes ("clear CTAs") and misses ("nested menus hide admission info").

🚩 Key Takeaways

✔ **IA is strategic:** taxonomy choices alter business outcomes.
✔ **Garrett's planes** stop premature pixel-pushing.
✔ **Navigation = cognitive GPS**—neglect it and users bail.
✔ **Metadata & vocabularies** are the unsung heroes of findability.
✔ **User research** (personas, PACT, usability tests) is your design insurance policy.

> *"Minimize the mental costs of sense-making, decision-making, remembering and planning."* – Brinck et al.

🌟 **Next step:** conduct a **content audit** of any site you love/hate; classify each page by genre, IA scheme, and navigation quality. You'll *feel* the lecture come alive.

Lecture 17: Social Media

---

1. Overview of the Lecture

Lecture 17 situates *Social Media* within the broader trajectory of interactive-system design. It:

- Traces the **technical origins** of the Web, *Web 2.0*, and the "social turn."
- Unpacks the **sociotechnical forces**—participation, crowdsourcing, recommender engines—that distinguish social platforms from earlier "read-only" websites.
- Surveys the **current landscape** (top platforms, business uses, UX patterns) and anticipates **future developments** such as location-based apps, gamification, cloud computing, and IoT.

> "Social media is a form of digital communication that allows users to form online networks and communities for socializing, sharing information, and posting user-created content."

2. What *Is* Social Media?

Social media comprises *networked, user-generated communication channels* that operate atop Internet protocols and browser-based (or mobile) clients.
Key traits 🌐:

Trait	Design Implication
User-generated content	Interfaces must support low-friction creation (camera upload, WYSIWYG editors).
Networked publics	Privacy controls, visibility settings, and algorithmic curation become core IA problems.
Persistent profiles & activity trails	Data models store identity, edges, and interaction events; designers must visualize “liveness.”
Real-time feedback loops	Notification systems, badge-based gamification, *infinite scroll* patterns.

Examples: personal *Stories* on Instagram, *Duets* on TikTok, community *Subreddits*—each a specialized affordance layered on the same underlying principles.

---

3. Origins & Evolution

3.1 Pre-Web Seeds

- **Bulletin Board Systems (BBS)** and **Usenet** provided threaded conversation models in the 1980s.
- **MUDs (Multi-User Dungeons)** pioneered identity play and synchronous chat—precursors to today’s avatars in Roblox.

3.2 Web 1.0 → Web 2.0

- 1989 – Tim Berners-Lee proposes the **World Wide Web** at CERN. Hypertext *documents* link via HTTP.
- Pre-1993 access required arcane command syntax; “ordinary users” arrived only after graphic browsers (Mosaic, 1993).
- Around 2004, O’Reilly’s **Web 2.0** reframed the Web as a *platform*:

> “Web 2.0 is about participation more than publishing; ordinary people… supply the content and the trail of their activities adds value.”

- **APIs** expose functionality (Flickr, Google Maps mash-ups).
- New revenue logics: ad auctions, data brokerage, freemium SaaS.

3.3 First “Social Networks”

- **Friendster (2002)** and **MySpace (2003)** formalised profile-centric interaction; MySpace hit *one million monthly actives* in 2004.
- **Facebook (2004)** pivoted from campus directory to global identity layer—introducing the News Feed (2006) that normalised algorithmic aggregation.

- **Twitter → X (2006)** distilled status updates into 140-character micro-blogs, spawning real-time hashtag publics.

---

4. From Participation to Platform Power

The early exuberance ("*Internet time replaced reality*" during the dot-com boom → crash) demonstrates how technical affordances intertwine with capital cycles.

- **Crowdsourcing** (Howe, 2006) leverages distributed labour—e.g., Wikipedia's *prosumer* model.
- **Collective intelligence** becomes both a *design resource* (implicit tagging, social navigation trails) and a *business moat* (network effects).
- **Regulatory lens**: 2025 antitrust cases against Meta argue that "social networking" can't be isolated when TikTok, iMessage, Discord blur product boundaries. (Mark Zuckerberg Says Social Media Is Over)

---

5. Background Ideas for Designing Social Systems

Concept	Core Mechanism	Example UX Pattern
Social Navigation	Render other users' presence/actions to guide newcomers.	Activity heat-maps in GitHub commit graphs.
Recommender Systems	Collaborative or content-based filtering clusters users/items by similarity.	"Customers who bought *Sapiens* also bought *Homo Deus*."
Social Translucence	Visibility + Awareness + Accountability; show but do not overwhelm.	Slack "typing…" indicator, Babble marble proxy.
History-Enriched Environments ('readware')	Use temporal traces (visited links change colour) to signal prior exploration.	Purple vs. blue hyperlinks in browsers.

---

6. Contemporary Platform Ecology

6.1 User-Scale Numbers (Feb 2025) 📈

Rank	Platform	MAUs (billions)
1	Facebook	**3.07** (Global Social Media Statistics - DataReportal)
2	YouTube	2.54
3	WhatsApp	2.00
4	Instagram	2.00
5	TikTok	1.59
6	WeChat	1.39
7	Telegram	0.95

Rank	Platform	MAUs (billions)
8	Messenger	0.947
9	Snapchat	0.85

Slide figures align with independent DataReportal & Statista datasets. (Most popular social networks worldwide as of February 2025)

6.2 Usage Patterns

- **Average time/day: 2 h 21 m** across ~6.8 platforms.
- **Business integration:** Targeted ads, social commerce (*Facebook Marketplace*), influencer sponsorships.

6.3 Professional & Civic Use Cases

Actors (fan engagement), NGOs (fund-raising), politicians (real-time policy messaging), public-safety agencies (emergency alerts).

---

7. Sharing, Tagging & Folksonomies

- **Tags** serve as user-generated metadata enabling *folksonomy* navigation (e.g., #Foodie in Instagram).
- **Del.icio.us** pioneered social bookmarking; **Digg** introduced crowd-voted news ranking.
- **Tag clouds** visualise frequency; larger font ⇢ higher occurrence, supporting quick gestalt scanning.

> “A tag cloud is a visual representation of the most popular words found in free-form text.”

Design concern: avoid “tag spam” ➜ implement trust signals (reputation scores, down-votes).

---

8. The Developing Web: From Screens to Contexts

8.1 Location-Based Services & Gamification 🎯

- GPS-enabled phones let apps like Pokémon GO overlay game mechanics on real-world coordinates.
- Health apps (e.g., Strava) add badges & leaderboards to motivate behaviour change.

8.2 Cloud Computing & “Dumb Terminals”

- **Elastic Compute Cloud (EC2)** exemplifies pay-per-use infrastructure: developers rent cycles; users access documents anywhere (e.g., Google Docs).
- UX impact: persistent session state, cross-device sync, expectation of *zero-install* apps.

8.3 Internet of Things (IoT)

- Social paradigms extend to objects: a *smart fridge* tweets when milk runs low; wearables auto-share workout data.
- Designers must balance convenience with privacy—contextual integrity over blanket consent.

---

9. Class Activity → Design Analysis Framework

The lecture ends by challenging students to:

1. **Pick a current platform** (e.g., TikTok).
2. **Analyse**
   - Interaction design (gesture loops, Fitts' Law on swipe regions).
   - Visual language (bold sans-serif type, trend colours).
   - Tech stack (ByteDance's recommender micro-services, on-device inference).
   - Information architecture (For You vs. Following feeds).
3. **Propose enhancements** grounded in usability heuristics (e.g., Nielsen) *and* ethical design (e.g., minimise doom-scroll loops).

> "Explain what changes should be made in these platforms to enhance the user experience and usability."

---

10. Conclusion & Key Takeaways

✔ **Social media** repurposed the Web from a document repository to a *participatory, data-fueled social fabric*.
✔ Design principles evolved: social navigation, recommender engines, visibility/awareness loops.
✔ Business and civic life now operate inside these networks; user experience decisions have macro-economic and political consequences.
✔ The horizon points to *context-rich* computing—location, cloud, IoT—demanding designers embed ethics, privacy, and inclusivity at every layer.

> "The Web is not a dead information space; it is a lively space where the user can see other shoppers moving around, consult specialist agents, and 'talk to' personnel."

🌟 *Remember: great social-media design is **people-first**—technology merely amplifies (or distorts) the sociability already wired into us.*

Lecture 18: Collaborative Environments 🤝

*(Part III of **Design of Interactive Systems** – Contexts for Designing Interactive Systems)*

---

1. Deep-Dive Overview

The lecture positions *collaborative environments*—often labelled **Computer-Supported Cooperative Work (CSCW)**—as socio-technical ecosystems purpose-built to let **people work together seamlessly across space, time and media**. It frames collaboration as more than a technical overlay; rather, it is a **blend of individual interaction, social dynamics, workflow orchestration, and the surrounding physical or virtual setting**.

> "Collaborative environments comprise **spaces and software designed to support people working together**."

---

2. Foundational Aims

1. **Understand the issues that make—or break—collaboration.**
2. **Explore the spectrum of technological support**, from shared diaries to multi-user VR.
3. **Grasp the nature of Collaborative Virtual Environments (CVEs)** and the design decisions unique to them.

These aims translate into a design mantra: *augment human communication while safeguarding motivation, privacy, and situational awareness.*

---

3. The Four-Factor Design Lens

When crafting collaborative spaces, the lecture insists that designers inspect four intertwined layers:

Design Focus	Core Questions to Probe	Illustrative Example
Individual Interaction	*Is each participant's toolset intuitive & empowering?*	A hinge-click gesture on a tablet whiteboard that instantly snaps a sticky note.
Social Interaction	*How are norms, trust, & etiquette manifested?*	Emoji reactions in Slack that show appreciation without cluttering threads.
Workflow	*How are tasks decomposed, delegated & reintegrated?*	A kanban board auto-linking pull-requests to user stories.
Physical / Virtual Environment	*Does spatial layout (or 3-D topology) cue proximity & context?*	A VR lecture hall where avatars cluster at tables, mirroring real seating.

(Adapted from lecture slide on "Blended interaction.")

---

4. Classic Problems in Cooperative Work

4.1 The *Benefit–Burden* Asymmetry

> "The disparity between **who does the work and who gets the benefit**."

- **Pain-point:** Contributors must keep a shared calendar updated, yet only meeting organisers directly gain.
- **Design Remedy:** Automate data capture (e.g., email-to-calendar parsing) so effort & benefit realign.

4.2 *Critical Mass*

Groupware grows valuable only after enough users cross a **participation threshold**. A half-empty bulletin board breeds abandonment.

4.3 Social & Privacy Tensions

Moving a personal diary online converts a once-private artifact into a quasi-public resource. Designers must **surface permissioning granularity** and adopt a principle of *progressive disclosure*.

4.4 The *Space–Time* Matrix

Building on DeSanctis & Gallupe (1987), collaboration may be:

	Same Time	**Different Time**
Same Place	Brainstorm around a touchscreen table	Sticky notes left on a project wall
Different Place	Live Zoom design critique	Git commits & pull-requests

The matrix is more than taxonomy—it surfaces latency, awareness, and bandwidth constraints that dictate tool affordances.

---

5. *Articulation* & *Awareness*—The Twin Pillars

Concept	Essence	Design Imperatives
Articulation	Breaking the work into sub-tasks, delegating, then re-integrating.	Provide **task granularity controls**, versioning, and traceable ownership.
Awareness	Knowing *who* is doing *what*, *where*, and *when*.	Lightweight cues (tele-pointers, presence dots) plus unobtrusive update streams.

> "In distributed environments designers need to ensure that **collaborators are aware of changes** that happen."

🔑 *Trade-off*: **Privacy vs Awareness** and **Awareness vs Disruption**—over-notification can be as harmful as opacity. The Portholes system's minute-interval snapshots embodied this balancing act.

---

6. The Technology Arsenal

6.1 Communication Backbones

- **Video/Audio Conferencing** (Zoom, Meet, Teams) ⇒ Synchronous cue-rich exchanges.
- **Live Chat** ⇒ One-to-many or many-to-many textual bursts; ideal for quick consensus.

6.2 Shared Workspaces

- **Content Management Portals** (SharePoint, Confluence).
- **Threaded Bulletin Boards**—asynchronous idea incubation.

6.3 Shared & Electronic Whiteboards

> "Users are normally represented as **tele-pointers which are colour-coded or labelled**."

Design nuances:

- **Latency tolerance** must be < 100 ms to feel co-present.
- **Gestural parity**—physical pen rotations should map to digital stroke width.

6.4 Groupware Toolkits

GroupKit demonstrates how reusable network, session-control, and concurrency primitives can **slash development effort for multi-user editors**.

6.5 Awareness Applications

- **Portholes** (periodic photo thumbnails) → fosters ambient awareness with minimal bandwidth.
- Modern analogues: Slack *"Active"* badges; Google Docs cursor flags.

6.6 *Roomware*

> "Integration of furniture, doors, walls … to facilitate interaction."

Imagine walls doubling as touch displays, tables embedding NFC readers, doors logging occupancy—all orchestrated into a **sensor-rich collaboration habitat**.

6.7 Collaborative Virtual Environments (CVEs)

Second Life-style 3-D worlds where avatars **co-locate in cyberspace**. Key design pivots include:

1. **Spatial audio** for directional cues.
2. **Embodied avatars** for gestural nuance.
3. **Physics engines** to let virtual objects behave credibly.

---

7. Evaluation & Success Metrics 📊

The lecture advocates multi-layer assessment:

Artefact to Evaluate	Sample Metric	Why It Matters
Intermediate Reports	Alignment with articulation plan	Catches drift early.
Usability Tests	Task-completion time & error rate	Quantifies friction points.
Awareness Efficacy	Recall of teammates' actions	Gauges ambient information flow.
Code Quality	Cyclomatic complexity, test coverage	Ensures long-term maintainability.
Real-Time Demos	Latency under load	Validates scalability.

Extra credit arises from **novelty, workshop presentations, and live deployments**—a nudge toward research-grade outcomes.

---

8. Class Activities & Reflective Prompts

1. **Identify non-tech awareness cues** (e.g., coffee mug on desk → colleague present).
2. **Catalogue your own collaboration stack**; map each tool to issues (critical mass, privacy, articulation).
   These exercises ground theory in lived experience, reinforcing that **design decisions reverberate through social practice**.

Concluding Insights ✨

- **Collaboration is a choreography**—balancing personal autonomy with group coherence.
- **Designers wield leverage** at four layers: personal UX, social contracts, workflow structure, spatial/virtual affordances.
- **Awareness & articulation** are interdependent: without awareness, articulation fragments; without articulation, awareness overloads.
- **Every technological intervention carries a privacy & disruption budget**—overspend at your peril.
- **Future horizons** (CVEs, roomware, AI-mediated facilitation) will demand ever more nuanced orchestration of human factors and technical infrastructure.

> "Introducing collaborative environments can **disrupt the balance between private and public spaces**."

Design boldly—but always with *people* at the nucleus.

Lecture 19: Agents & Avatars 🧭

---

1. Position of the Lecture in DIS

Lecture 19 sits in **Part III – "Contexts for Designing Interactive Systems."** It follows the web-, social-media-, and collaboration-centric chapters (14–16) and asks us to look beyond static interfaces toward **software entities that sense, decide, and act.**

---

2. Defining an Agent—Beyond Ordinary Code 🤖

> *"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators."*

Agent Type	Sensors	Actuators	Everyday Example
Human	Eyes, ears, skin	Hands, voice	You reading & typing
Robot	Cameras, LiDAR, IR	Motors, grippers	Warehouse Kiva bot
Software	Keystrokes, packets, files	Screen output, API calls	Email-filter plugin

The definition highlights **autonomy** and **situatedness**: an agent chooses *what* to do (*autonomy*) and its choice depends on a **changing, partially observable world** (*situatedness*).

---

3. Aims of the Lecture

- **Describe key features** of interface agents (autonomy, reactivity, pro-activity, social ability).
- **Explain the conceptual model** of agents—including *person*, *domain*, and *interaction* models.
- **Introduce user modelling** as the backbone of adaptation.
- **Showcase real systems** that employ agents or avatars.

---

4. Strong vs. Weak Views of Agents

View	Capabilities	Illustrative Quote
Strong (BDI)	"Beliefs, desires, intentions; can plan, learn, adapt."	*"They have beliefs, desires and intentions and can…communicate."*
Weak (Interface Helper)	Task-specific heuristics, limited learning	Clippy's ancestor in MS Office

Strong agents resemble **goal-directed colleagues**, weak ones resemble **clever tools**.

---

5. How Agents Help Humans 💡

- **Guides** – explain an information space (e.g., museum robot).
- **Reminders** – manage calendars, deadlines.
- **Monitors** – watch mailing lists for relevant posts.
- **Collaborators** – co-create (e.g., pair-programming AI).
- **Surrogates** – stand in during meetings (AI note-taker).

---

6. Two Fundamental Agent Species

1. **Personal Agents** – know an *individual* deeply (habits, preferences).
2. **Domain Agents** – know a *task* deeply (indexing, spell-checking).

Hybrid designs (e.g., Siri) mash both: *your* music taste **and** the *music domain*.

---

7. Robots: Physical Embodiments of Agents

- **Industrial**: pre-programmed car-assembly arms or autonomous patrol drones.
- **Domestic**: 🏡 Roomba vacuums, lawn-mowing bots—each a mobile agent with embedded sensors/actuators.

---

8. Agents as Adaptive Systems 🔄

Systems nest (sub- & super-systems) and interact at three cognitive depths:

Level	What agent must "know"	Travel Example	Hammer Example
Physical	Clear audio, readable UI	Loudspeaker volume	Mass & handle grip
Conceptual	Flights, gates, language	"LH760 → Gate A3"	Nail types
Intentional	User purpose, goal	Catching a Delhi flight	Building a chair

Types of adaptivity (Browne et al., 1990):

1. **Reflexive** – rule-based (thermostat).

2. **Sequential Memory** – predictive text learns from earlier keystrokes.
3. **Trial-and-Error** – game-playing AI evaluates outcomes before adjusting.

---

9. Canonical Architecture for Interface Agents 🏗

```
┌───────────┐      ┌──────────┐      ┌──────────────────┐
│ Person    │ ---> │ Dialogue │ ---> │ Knowledge Base   │
│  Model    │      │  Record  │      │ (rules, ML, etc.)│
└───────────┘      └──────────┘      └──────────────────┘
      ↑                 ↓                     ↑
      └────── Domain Model & Context ─────────┘
```

1. **Person Model** – habits, skills, goals 📚.
2. **Domain Model** – objects & intents of the task.
3. **Interaction Model** – dialogue record (keystrokes, gaze, skin conductance…) and the *reasoning* that drives adaptation.

---

10. Data the Dialogue Record May Store

- Keystroke streams & mouse paths
- Facial expressions (via webcam)
- Eye-tracking vectors
- Speech prosody & transcript
- Physiological signals (GSR, grip)
  Why? To build a **rich context state** that fuels just-in-time adaptations (e.g., offering help when gaze stalls).

---

11. Application Domains

11.1 Natural-Language Agents 🗣

Dream since Turing: a system that "understands." Modern chatbots use **NLU pipelines** → intent, slots → actions. Ambiguity requires user modelling to guess *focus of attention*.

11.2 Intelligent Tutoring Systems (ITS) 🎓

- Maintain a *student model*: concepts mastered, misconceptions.
- Decide *when* to intervene, *what* hint to give.
- Aim to reduce variance found in human instruction quality.

---

12. Avatars & Conversational Agents 🧑‍💻

An **avatar** visually embodies the agent—2-D sprite, 3-D character, or tangible toy.

- **Nabaztag**: IoT bunny with LED ears; reads email via TTS.

- **Ananova**: virtual newsreader; early lip-sync attempts show importance of prosody.

---

13. Companions: From Interaction to Relationship ❤

> *"Change interactions into relationships."*

Design axes:

Axis	"Pet" End	"Assistant" End
Utility	AIBO dog—play, zero tasks	Carebot—dispenses meds
Form	Cartoon 2-D blob	Photo-real 3-D humanoid
Emotion	Mood mirroring, self-disclosure	Professional empathy

Emotion matters – attractive agents boost user creativity (Norman). **Personality fit** increases trust: assertive users prefer assertive computers (Reeves & Nass 1996).

Bickmore & Picard (2005) stress **appraisal support** and gentle *persuasion* (e.g., fitness coach nudging harder runs).

---

14. Companion Architecture in Practice

1. **Multimodal input fusion** – touch, ASR, sensor signals.
2. **Emotion detection** – voice tone + sentiment analysis.
3. **Dialogue understanding** – NLU + emotion + entity context.
4. **Decision & Planning** – consult domain + user models.
5. **Multimodal output** – speech text, intonation, avatar gestures.

---

15. Critical Design Reflections

- **Privacy & Ethics**: sensors record intimate cues—need informed consent & data minimization.
- **Explainability**: adaptive moves must be *legible* or users lose trust.
- **Persuasive Tech Dangers**: nudge can slip into manipulation—designers must define ethical boundaries.

---

16. Key Takeaways 🌟

✔ **Agents sense, decide, & act**—moving HCI from passive tools to active teammates.
✔ **Person-Domain-Interaction models** organize agent knowledge.
✔ **Adaptivity levels** range from thermostat rules to BDI reasoning.
✔ **Avatars & companions** blend utility with emotion, forging long-term relationships.
✔ **Trust, personality match, and transparency** are non-negotiable for adoption.

> In short, Lecture 19 invites designers to treat *interfaces as living partners*—autonomous, context-aware, and emotionally attuned. The challenge: give them **just enough smarts** to help without over-stepping the human in the loop. 🧑‍🚀