Higher Education and Practice Vol. 2 No. 10, 2025

Application of Component Construction Thinking in UI Design Courses

Mengyue Zheng
School of Humanities, Xiamen Huaxia University, Xiamen, China

Abstract: In today's UI design field, component libraries have become an indispensable tool that significantly improves design efficiency. They not only simplify the design process but also establish a solid bridge between design specifications and development implementation. Currently, there is a significant gap between talent cultivation and job requirements in UI design course teaching at universities. This paper aims to explore the basic principles and importance of component libraries, as well as how to effectively cultivate component thinking in UI design courses for undergraduate students majoring in Visual Communication Design. Furthermore, it analyzes the design principles and practical applications of component libraries, and establishes a teaching experimental framework for UI design component thinking, hoping to build a closer connection between the training mode of UI design talents and job demands.

Keywords: UI Design; Components; Constructive Thinking; Course Design

1. Concept of Component Libraries Based on Atomic Methodology

1.1 Concept of Component Libraries

A component library is a standardized collection of components constructed by organizing and summarizing universal interface elements (i.e., controls). This collection aims to realize rapid reuse and batch modification of components, thereby significantly improving design efficiency. As a powerful toolset, component libraries provide clear and standardized guidelines and specifications, enabling designers to call components anytime and anywhere, greatly promoting collaboration among development teams, and ensuring the systematicness and consistency of user experience.

1.2 Principles of Atomic Methodology

As early as 2013, front-end developer Brad Forst proposed a compelling analogy in his book Atomic Design: "In the chemical world, everything is composed of atoms; atoms combine to form molecules, molecules form organisms, and these organisms ultimately constitute everything in the universe we know." He suggested that design components should be constructed according to five hierarchical levels: atoms (basic elements), molecules (combined elements), controls (functional units), templates (page layouts), and pages (complete interfaces), which ultimately form a complete and coordinated product interface. In the development of large-scale software projects, componentization has become a consensus—it not only improves development efficiency but also reduces maintenance costs [1]. He believed that this atomic methodology is also applicable to UI design.

When constructing complete interface components, their components can be understood according to the following levels:

Atoms: The smallest basic and indivisible elements that constitute the interface. **Molecules:** Components formed by combining two or more atoms, with clear functional or informational attributes. **Controls:** Modules with complete functions formed by combining multiple different molecules. **Templates:** Frameworks obtained by connecting atoms, molecules, and controls in parallel or series. **Pages:** Complete wireframes or high-fidelity interfaces formed by filling real content and refining details based on templates.

These pages are what users ultimately see, providing rich information and interactive experiences.

2. Analysis of the Current Situation of UI Design Courses and Curriculum Systems in Universities

2.1 Current Situation of UI Design Courses in

Universities

An increasing number of comprehensive universities and art academies have launched UI design courses to meet the growing industry demand, with these institutions mainly distributed in first-tier cities and economically developed regions. UI design courses are usually incorporated into related majors such as Digital Media Art Design, Visual Communication Design, and Interaction Design. In terms of course positioning, comprehensive universities and art academies focus on imparting basic theories and design skills, while application-oriented universities and vocational schools emphasize practical application and professional software proficiency training [2].

Due to the uniqueness of UI design, it is necessary to integrate theory with practice to enable students to apply design principles and methods to practical design projects. Therefore, the content and structure of UI design courses usually include: **Basic courses:** Covering design fundamentals, color theory, typography principles, visual communication principles, etc., to lay a solid theoretical foundation for students. Professional skills courses: Including the use of design software, user experience design, interaction design, etc. Practical project courses: Many institutions set up practical project links in UI design courses, allowing students to participate in the design and implementation of real projects to improve their practical abilities and teamwork skills. Some institutions have established cooperative relationships with enterprises to provide internships for students, enabling them to practice skills and accumulate experience in actual work.

Course development trends: Future UI design courses will pay more attention to practical teaching and innovative ability training, emphasize cooperation and communication with enterprises and industries, and focus on the requirements of job functions for practitioners to improve the pertinence and practicality of courses. At the same time, they will also pay more attention to the development of new technologies and trends, updating course content and teaching methods with the changes of the times and technological iterations.

2.2 Framework of UI Design Curriculum System

Mobile UI design courses mainly cover information design, interaction design, and visual design. Usually offered in the second or third year of undergraduate studies as a core course for Visual Communication Design majors, UI design courses are preceded by prerequisite courses such as layout design, software courses, and icon design, and also serve as an important connection for subsequent courses such as motion design and web design. Combining years of design and teaching experience, the author believes that a reasonable sequence and content of UI design courses should be: Overview of UI Design, Market Research, User Research, Competitive Product Analysis, Key Function Extraction, Product Information Architecture Design, Interactive Prototype Design, Interface Visual Design, Effect Diagram Output, Development, Testing, Launch, and Maintenance (Figure 1). This paper focuses on exploring the integration of component design thinking and methods into interface interaction design and visual design.

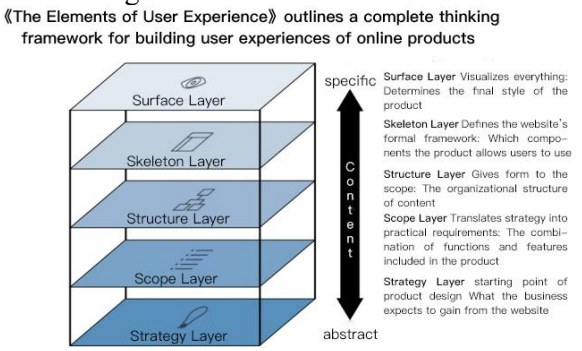

Figure 1. Workflow Based on the Five Elements of User Experience

3. Application of Component Libraries in UI Design Courses

3.1 Composition of Element Components

Each product platform usually has its own unique design system, including self-defined components and component libraries. Regardless of the platform to be adapted to, the primary step in learning components is to understand their types, which are generally divided into two categories: basic components and business components (also known as advanced components). Basic components, as underlying building units, are relatively independent with single functions. In contrast, business components are large-scale components composed of basic components, hence also called advanced components [3]. This paper focuses on exploring the construction thinking of basic components and their practical applications

in UI design courses. By deeply understanding the construction principles and usage methods of basic components, designers can complete design tasks more efficiently while ensuring design consistency and maintainability. There are four interface elements that affect basic components:

3.1.1 Interface color

Color plays a crucial role in product design style. It not only shapes a visual image consistent with the product tone but also effectively sorts out the relationship between various information levels in the interface, enriches visual hierarchy, and achieves visual harmony and balance [4]. In the design process, it is first necessary to define all colors used according to functional attributes, including primary colors, secondary colors, neutral colors, accent colors, and functional colors. Based on these basic colors, gradients and color shades can be further refined to meet different design needs. For the convenience of subsequent use and calling, these colors need to be grouped, classified, and given corresponding names. Through such a color management strategy, design teams can carry out product design more efficiently, ensure the appropriate use of colors in the interface, and bring users a high-quality visual experience.

3.1.2 Interface text

In the design of text styles in the interface, designers need to comprehensively consider four core elements: font, font size, font weight, and line height. To meet the needs of different usage scenarios, corresponding specifications and gradients need to be created for these elements. Text styles should be accompanied by clear usage instructions for quick and accurate calling in different scenarios.

3.1.3 System icons

In the process of icon design and drawing, the grid system plays a vital role in precisely controlling the size of icons of different shapes. Generally, elements such as active areas, decorative areas, and key line shapes are used to standardize the component structure of icons. Once the icons are drawn and finalized, it is necessary to define their common states, such as normal, selected, and disabled. Such standardized design not only ensures the clarity and consistency of icons but also helps the development team efficiently and accurately implement various states of icons in the subsequent development process [4]. During development, the development team can flexibly switch between different icon modes according to high-fidelity effect diagrams to ensure that icon elements in applications or interfaces present the best visual effect.

3.1.4 Basic components

Referring to the aforementioned atomic methodology, independent elements such as atoms and molecules are organized into various components (e.g., buttons, pop-ups, forms, option controls). These components are then combined, nested layer by layer, refined, sorted, and classified to ultimately form basic components.

3.2 Roles of Component Libraries in Interface Design

3.2.1 Maintaining systematicness and consistency

In product design, the transmission of each function or information needs to be closely combined with different usage scenarios. Due to the diversity of these scenarios, their visual expressions will naturally present various differences. However, with the specifications and constraints of component libraries, design and development teams can carry out design work in accordance with unified standards and rules within a preset framework. This not only ensures consistent quality of the output design works but also brings users a coherent and consistent experience.

3.2.2 Improving team collaboration efficiency

When multiple interfaces in a product use the same elements or components, the design team can directly call basic components from the component library to reduce repetitive design. If there is a need for unified modification of a component, modifying it individually in the component library will allow the shared components to be updated synchronously, achieving a significant improvement in efficiency. For the entire team, having a mature component library as a design reference enables the workflow to get on track quickly with minimal communication costs and trial-and-error costs.

3.2.3 Consolidating product brand image

Key elements in a product's brand genes, such as colors, fonts, and icon styles, all constitute important parts of the component library. By implementing consistent visual styles through componentization, designers can ensure the unity of the product design style, thereby creating a distinct and unique brand impression for users.

This consistency not only helps users better remember the product but also triggers emotional resonance and attachment, ultimately prompting users to form usage habits.

3.2.4 Complying with user usage habits

Through consistent design performance, componentized design can effectively maintain users' formed behavioral habits and thinking patterns, reducing unnecessary cognitive burdens during operation. In the design process, interface layouts and interactive operations should try to avoid going against users' behavioral patterns and thinking inertia to ensure that users can interact with the product smoothly and naturally. However, if attempting to highlight the product's unique design sense through reasonable differentiated design, it is necessary to weigh carefully and be prepared to bear the potential costs of user trial and error [5]. Such design decisions need to strike a balance between the innovation and stability of user experience.

3.2.5 Optimizing component upgrade and reuse

In the stage of component reuse, it is necessary to design a good component management method, such as using folders and configuration data files to organize component content and configurations. Provide a component library directory file listing all component information for designers to access and use. Continuously iterate and update the UI component library based on reuse feedback and usage to optimize component reuse performance and user experience. In UI design courses, components can be developed in small groups, and then a continuously improved and expanded component system can be established at the class or even major level. Updating after each course can form a mature component library.

3.3 Attributes of Components

3.3.1 Component shape

Design teams can quickly distinguish different types of components through shape differences. In design, the visual shape of components is usually defined by combining text and graphics. When designing components, it is necessary to combine visual elements such as shapes, colors, icons, and text, and reasonably arrange the hierarchical structure of components. Therefore, in the design process, it is necessary to fully consider the usage scenarios and requirements, and have a clear understanding of the shape of the components needed. Directly matching the component shape to the scenario will make the entire design process clear and unambiguous.

3.3.2 Interactive behavior

Behavior is related to interaction logic and business logic, telling users the results after operation. It is used to inform users of immediate feedback when triggering a component or the current state of the component.

3.3.3 Component states

Changes in component states can inform users of the current operation step, helping them understand the component's status. Common component states include: normal state, focused state, hover state, active state, loading state, and disabled state. The above component states need to be included in the component library in a standardized manner.

3.3.4 Context

The usage of components varies according to their scenarios or environments. This means that even the same component may need to be used differently in different contexts. In design, all elements are relative; the purpose of a component is not fixed but is defined and determined by various factors such as its layout position on the page, other component elements used with it, and the user's specific usage purposes. Such flexibility enables components to better adapt to various scenarios and needs, improving user experience.

3.4 Three Types of Components

Components can be roughly divided into three types: navigation components (for navigating information), input components (for users to input information), and information components (for delivering information to users).

Navigation components, used for navigation prompts or information display, include cards, lists, grids, carousels, tabs, menus, etc. Input components are used for inputting information or making selections. Currently, input components in many component libraries have similar shapes, designed to consider user habits and avoid unfamiliar components. Information components are used to convey information, delivering user selections according to the type of information to be input.

4. Application of Component Library Design Systems in UI Design Courses

In UI design courses for undergraduate Visual Communication Design majors, the author takes the development and design of mobile APP

products as the main teaching content. In the interface visual design part, component library thinking is integrated into students' learning and design practice, attempting a new teaching mode more adapted to job requirements to connect university education with design job needs.

4.1 Component Library Construction Thinking

Based on years of design and teaching experience, the author organizes and defines component-related content and knowledge as component management, including component style definition, component and component library design, construction, and iterative optimization.

When constructing components and component libraries, it is first necessary to follow and focus on the core design principles, concepts, and goals. Taking Apple's Human Interface Guidelines as an example, the three core tenets of system design mentioned include clarity, compliance, and hierarchy, as well as six principles: completeness, consistency, directness, feedback, metaphor, and control. These principles ensure that the design is clear, intuitive, and easy to understand, thereby improving user experience. Similarly, Android's Material Design 1 & 2 also proposes three design principles: "Material is metaphor", "Bold, graphic, intentional", and "Motion provides meaning". These principles emphasize the naturalness, intuitiveness, and dynamics of design, bringing users a more vivid and interesting experience. Domestically, Ant Design is an authoritative representative in the field of B-end design. Starting from design values, it extends corresponding design principles and further explores design patterns. This value-led design method ensures design consistency and coherence while reflecting in-depth understanding and care for user needs.

In summary, both the design of components and component libraries, as well as the construction of the overall design system, need to be guided by design principles, concepts, and goals to ensure design rationality, effectiveness, and user experience optimization.

4.2 Process of Component Library Design System

4.2.1 Understanding product structure

Understanding the product structure helps designers quickly build the basic framework of the component library, providing a basis for component classification and weight ranking. Secondly, different functional attributes have a significant impact on interface layout, but the structural layouts of the same business and functions are basically similar with high component reusability. It is not that designers do not want to make differences, but in the same industry, the same business attributes have formed a relatively mature structural layout for most users. Major changes will go against users' common habits, leading to user dissatisfaction and not worth the loss. Therefore, by understanding the product structure, more differences can be placed on component details, which are more acceptable to users.

4.2.2 Component organization and classification

At the UI interface level, components can be divided into four types: native components, extended components, custom components, and encapsulated components. Native components and extended components are system-built, classified as basic components. Custom components and encapsulated components have strong relevance to product functions, thus becoming attribute components. Clarifying these two definitions helps design and development teams reasonably plan the preliminary work of building component libraries and facilitates later calling:

- **Native components:** Components inherently provided by the system, such as buttons, pop-ups, and navigation bars.
- **Extended components:** Extended based on native components, such as adding icons to toast pop-ups or functional entrances to navigation bars [6].
- **Custom components:** Components designed with product characteristics ignoring system-built components, such as product lists.
- **Encapsulated components:** Common complex components combined and encapsulated based on frequently used product components, such as calendar components.

4.2.3 Component structure segmentation

Structure segmentation involves deconstructing independent components into the smallest atomic elements, fully improving the reusability of small components. When modifications are needed, independent adjustments can be made with global responses, followed by recombination to allow alternating changes of different modules. Repeated use of segmentation and recombination methods can multiply the

final number and styles of components (Figure 2).

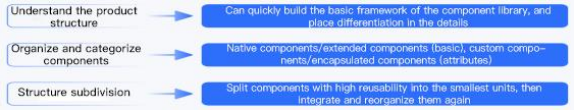

Figure 2. Component Structure Segmentation Table

4.3 Design System of Component Libraries in Courses

A design system is also called a design system. Component systems and design languages define what products and designs should be but do not answer how to achieve them. These questions are addressed by the design system. A design system usually includes a collection of the following contents: design values, design principles, component systems, style guides, best practices, tool resources, and workflows (SOP), etc. (Figure 3).

Figure 3. Component Design System

An effective design system can help teams improve design decisions, optimize the workflow of design and development, and reduce the risk of errors. Establishing a design system is also a part of team management, which helps to provide guidance to all participating members while ensuring that the team's work does not break due to the absence of a key member.

Applying component thinking in UI design courses is a very important teaching method that helps students better understand the modularity, reusability, and maintainability of design. The following are specific methods for applying component thinking in UI design courses:

4.3.1 Introducing the concept of component thinking

At the beginning of the course, introduce the concept of component thinking to students and explain why it is crucial for UI design. Illustrate the practical application of component thinking in UI design with examples such as the reuse of common elements like website navigation bars, buttons, and forms.

4.3.2 Teaching component creation

Guide students to identify and extract reusable design elements and convert them into components. Teach students how to create and save components using design software. Emphasize component naming conventions to ensure the clarity and usability of the component library.

4.3.3 Component classification and organization

Guide students to classify components (e.g., buttons, input boxes, icons, layouts) for easy management and retrieval. Teach students how to organize component libraries, such as grouping by function, project, or theme.

4.3.4 Component reuse and modification

Emphasize component reusability, guiding students on how to reuse existing components in new design projects or similar products and modify component styles and attributes according to specific needs to adapt to different design scenarios.

4.3.5 Component application in project practice

In practical design projects, require students to apply component thinking, reusing existing components as much as possible. Guide students to create new components based on project needs and update them to the component library to form a more complete and expanded component system.

4.3.6 Component library sharing and maintenance

Encourage students to share their component libraries for mutual learning and exchange. After resource sharing among groups or teams, form a class or major-level component library system. In this process, teach students to maintain and update the component library to ensure its timeliness and availability.

4.3.7 Case analysis

Display successful UI design cases in the course, analyzing the application and effects of component thinking. Guide students to think about how to apply component thinking in their own designs to improve design efficiency and quality.

Through the above teaching methods, students can better master component thinking in UI design courses and apply it in practical design projects. This will help them improve design efficiency and quality, laying a solid foundation for future career development [7].

4.4 Judging the Aesthetics and Usability of Components

Students often ask what makes a component beautiful or how to define a component as good and appropriate. In fact, the concept of "beautiful design" is subjective and difficult to define absolutely. Based on industry evaluation standards, the author summarizes four design rules for design teams to verify and judge:

4.4.1 Form fits function

"Form follows function" is an early product design concept from the Bauhaus, which is also applicable to interaction design. The evaluation of "beauty" requires specific environments and content support. No matter how attractive the visual performance is, if it fails to convey a clear message or correspond to relevant functions, deviating from practical purposes, it cannot be called "beautiful".

4.4.2 Natural and comfortable operation

Humans are products of nature. Facing nature, most people feel happy and relaxed. Ant Design also points out in its design values that "beautiful" interaction reflects naturalness in two aspects:

- **Perceptual naturalness:** Elements such as layout, color, illustration, and icons in interface design should fully draw on natural laws to reduce user cognitive costs and bring a real and smooth experience.
- **Behavioral naturalness:** In the process of interacting with product interfaces, designers should fully understand the relationship between users and task goals, and organize system functions and services. Scenario-based design helps users make smooth decisions, reduce operational obstacles, save users' mental and physical efforts, and make human-computer interaction more natural and smooth [8]. This strengthens the concept of designing centered on user experience.

4.4.3 Positive and progressive thinking

Appropriate and beautiful designs embody positive values in specific things and should spread positive energy and ideological cognition through positive design methods [9]. Dangerous, aggressive, and unhealthy design expressions cannot be called beautiful.

4.4.4 Design keeping pace with the times

"Beauty" is the embodiment of social ideology at the visual level, constantly changing with the development of human history and culture. Therefore, conformity to current popular trends can also be a criterion for judging appropriate and beautiful designs. For example, current popular styles and forward-looking design expressions can endow products with more convincing visual performance.

In addition, the author believes that these four judgment criteria are progressive. After the most basic "fit function" is perfectly matched with functions, we can pursue "natural comfort", then "positively influence others", and finally "keep pace with the times" to continuously innovate following the general trends of the times. This is also a spiral upward process of design development [10].

5. Conclusion

Component libraries are powerful efficiency-enhancing tools. Fully understanding and rationally using them can reduce a lot of repetitive work. From design specifications to component libraries, and finally to development restoration, they build a new bridge between design and development. The establishment of component libraries provides a unified standard internally, making it easier for students to understand and master. While significantly improving work efficiency, consistency is also guaranteed, allowing designers and developers to focus more time on polishing product details and realizing design empowerment for products.

Acknowledgments

This work is supported by Fujian Province Education Science Planning Project FJJKB24-167.

References

[1] Han L, Chen M. Application of design component libraries in the field of UI design. Shanghai Packaging, 2023, (11): 169-171.
[2] Xue H J, Li J J. Career planning and innovative development of digital media talents from the perspective of AIGC. Modern Vocational Education, 2024, (10): 153-156.
[3] Han L, Chen M. Application of design component libraries in the field of UI design. Shanghai Packaging, 2023, (11): 169-171.
[4] Sui Y. Visual Design of Internet Products. Tsinghua University Press, 2015.
[5] Hu X C. Research on interaction design in the Chinese context. Science & Technology for Development, 2014, 6(02): 155-156+124.
[6] Chen Z G, Li S G. Research on 3D user interfaces in virtual environments. Packaging Engineering, 2010, 31(02): 37-40+44.

[7] Fang J L. Research on the application and development of UI interface design in the new media era. Journal of Beijing Institute of Graphic Communication, 2019, 27(12): 30-32.
[8] Wu Q. The domain and boundary of interaction design. Zhuangshi, 2010, (01): 34-37.
[9] Sun X T, Zheng T. Construction and reform of new media art courses under the background of innovation and entrepreneurship education. Education Modernization, 2017, 4(45): 75-76.
[10] Li Y Y. Injecting the "spirit, charm, and realm" of traditional Chinese culture and art into modern UI design. Journal of Beijing Institute of Graphic Communication, 2019, 27(08): 30-32.