

동적 인터랙션에 대한 디자이너와 개발자의 인식 비교 연구

카드소팅과 개념도를 활용한 인식 구조 분석

A Comparative Study of Designers' and Developers' Perceptions of Dynamic Interactions

An Analysis of Perceptual Structures Using Card Sorting and Concept Mapping

주 저 자 : 박선후 (Park, Sunhoo) 국민대학교 TED 스마트경험디자인학과 석사과정

공 동 저 자 : 이보영 (Lee, Boyeong) 국민대학교 TED 스마트경험디자인학과 석사과정

교 신 저 자 : 허정윤 (Heo, Jeongyun) 국민대학교 TED 스마트경험디자인학과 교수
yuniheo@kookmin.ac.kr

<https://doi.org/10.46248/kidrs.2026.2.358>

접수일 2026. 06. 10. / 심사완료일 2026. 06. 13. / 게재확정일 2026. 06. 15. / 게재일 2026. 06. 30.

Abstract

In the digital service environment, Dynamic interactions are garnering attention as a key differentiating factor in User Experience (UX). However, in design systems and component-centric collaboration environments, the design intent and experiential value of Dynamic Interactions are often not fully conveyed, and differences in interpretation between designers and developers regarding the same interaction elements can lead to issues in the collaboration process. To analyze the differences in perception between job roles regarding the delivery of Dynamic Interactions, this study conducted semi-structured interviews, card sorting, and concept mapping experiments with designers and developers. The analysis revealed that while designers understood Dynamic Interactions in terms of user experience and state changes, developers interpreted them in terms of implementable system structures, such as conditions, state values, and attribute values. In particular, although both groups recognized state changes as a core element, they differed in their understanding: designers perceived them as changes in user experience, while developers perceived them as transitions in system state. This study empirically identifies perceptual structures of Dynamic Interactions by job role and suggests the need for collaboration support measures to bridge designers' experiential intent and developers' implementation specifications. Therefore, this study aims to provide foundational data for the future design of AI-based collaboration support, based on the differences in the perception structures of designers and developers.

Keyword

디자이너-개발자 협업(Designer-Developer Collaboration), 마이크로인터랙션(Microinteraction), 인식차이(Perception Gap), 카드소팅(Card Sorting), 개념도(Concept Map)

요약

디지털 서비스 환경에서 동적 인터랙션은 사용자 경험(UX)의 중요한 차별화 요소로 주목받고 있다. 그러나 디자인 시스템과 컴포넌트 중심의 협업 환경에서는 동적 인터랙션의 설계 의도와 경험적 가치가 충분히 전달되지 않는 경우가 발생하며, 동일한 인터랙션 요소에 대한 디자이너와 개발자의 해석 차이는 협업 과정의 문제로 이어질 수 있다. 본 연구는 동적 인터랙션 전달 과정에서 나타나는 직군 간 인식 차이를 분석하기 위해 디자이너와 개발자를 대상으로 반구조화 인터뷰, 카드소팅, 개념도 실험을 수행하였다. 분석 결과, 디자이너는 사용자 경험과 상태 변화를 중심으로 동적 인터랙션을 이해한 반면, 개발자는 조건, 상태값, 속성값 등 구현 가능한 시스템 구조를 중심으로 해석하였다. 특히 두 직군 모두 상태 변화를 핵심 요소로 인식하였으나, 디자이너는 이를 사용자 경험의 변화로, 개발자는 시스템 상태의 전환으로 이해하는 차이를 보였다. 본 연구는 동적 인터랙션에 대한 직군별 인식 구조를 실증적으로 규명하였으며, 디자이너의 경험적 의도와 개발자의 구현 명세를 연결할 수 있는 협업 지원 방안의 필요성을 제시한다. 나아가 디자이너와 개발자의 인식 구조 차이를 바탕으로, 향후 AI 기반 협업 지원 설계를 위한 기초 자료를 제공하고자 한다.

목차

1. 서론

- 1-1. 연구 문제 및 배경
- 1-2. 연구 목적 및 질문

1-3. 연구 방법 및 범위

2. 이론적 배경 및 연구 프레임워크

- 2-1. 동적 인터랙션과 마이크로인터랙션

- 2-2. 사용자 가치와 구현
- 2-3. 디자이너-개발자 협업 과정의 멘탈모델의 중요성
- 2-4. 분석을 위한 개념적 프레임워크

3. 연구 방법

- 3-1. 연구 설계
- 3-2. 연구 대상 및 참여자
- 3-3. 카드범주의 도출 및 구성
- 3-4. 실험 절차
- 3-5. 분석 방법

4. 연구 결과

- 4-1. 카드 그룹화 결과
- 4-2. 카드 그룹 명명 결과
- 4-3. 중요도 인식 비교
- 4-4. 직군 간 인식 차이 종합

5. 논의 및 결론

참고문헌

1. 서론

1-1. 연구 문제 및 배경

최근 디지털 제품 개발 환경에서는 디자인 시스템, 컴포넌트 기반 설계, 그리고 인공지능 기반 개발 도구의 확산으로 인터페이스 설계와 구현의 효율성이 크게 향상되고 있다.¹⁾ 애플(Apple. Human Interface Guidelines, HIG),²⁾ 구글(Google. Material Design)³⁾과 같은 플랫폼 사업자들이 제공하는 설계 가이드라인과 디자인 시스템은 사용자 인터페이스 구성 요소의 표준화를 확산하였으며,⁴⁾ 재사용 가능한 컴포넌트와 디자인 시스템 라이브러리가 널리 공유되면서 서비스 간 UI 구조와 시각적 패턴 또한 점차 유사한 형태로 정착되고 있다.⁵⁾

이와 함께 인공지능 기반 코딩 도구와 바이브 코딩(Vibe Coding)과 같은 새로운 개발 방식은 인터페이스 구현 과정의 속도와 생산성을 향상시키고 있다.⁶⁾ 이를 통해 반복적인 UI 구현 작업에 소요되는 시간과 노력이 감소하면서, 개발 초기 단계에서도 화면 설계와 검증을 보다 신속하게 수행할 수 있는 환경이 마련되고 있다. 이러한 변화는 설계 및 구현 과정의 부담을 완화하는 동시에, 사용자 행동에 대한 반응, 시스템 상태 변화, 피드백, 전환 흐름과 같은 보다 세밀한 사용자 경험 요소에 관심을 집중할 수 있는 여건을 제공하고 있다.

한편, 인터페이스 구성 요소와 시각적 패턴이 점차 표준화됨에 따라 디지털 제품의 경쟁력은 더 이상 화면 구성이나 시각적 완성도에 의해서만 결정되지 않는다.⁷⁾ 사용자의 행동에 즉각적으로 반응하고 시스템 상태를 전달하는 동적 인터랙션의 품질이 사용자 경험의 핵심 차별화 요소로 부상하고 있으며,⁸⁾ 특히 버튼 피

Systems, Article 196, 2021, pp.1-14.

- 1) Xu, W., "Maximizing the value of enterprise human-computer interaction standards: Strategies and applications", arXiv, 2021, pp.1-16.
- 2) Human Interface Guidelines, Apple Inc., 2025. (2026.06.01.)
<https://developer.apple.com/design/human-interface-guidelines/>
- 3) Material Design 3, Google, 2025. (2026.06.01.)
<https://m3.material.io/>
- 4) Lamine, Y., & Cheng, J., "Understanding and supporting the design systems practice", arXiv, 2022, pp.1-26.
- 5) Goree, S., Doosti, B., Crandall, D., & Su, N. M., "Investigating the homogenization of web design: A mixed-methods approach", Proceedings of the 2021 CHI Conference on Human Factors in Computing

- 6) Chen, Y., Shi, C., & Chen, L., "SpecifyUI: Supporting iterative UI design intent expression through structured specifications and generative AI", arXiv, 2025, pp.1-27.
- 7) Lenz, E., Diefenbach, S., & Hassenzahl, M., "Aesthetics of interaction: a literature synthesis", Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, ACM, 2014, pp.628 - 637.
- 8) Makkan, N., Brosens, J., & Kruger, R., "Designing for positive emotional responses in users of

드백, 상태 변화, 로딩, 화면 전환과 같은 마이크로인터랙션은 이러한 경험 품질을 형성하는 중요한 설계 요소로 주목받고 있다.⁹⁾

동적 인터랙션(Dynamic Interaction)은 사용자의 행동에 반응하여 시스템 상태를 전달하고, 행동을 유도하며, 경험의 흐름을 형성하는 일련의 상호작용을 의미한다.¹⁰⁾¹¹⁾ 특히 마이크로인터랙션(Microinteraction)은 사용자의 행동에 대한 피드백, 상태 변화, 로딩 과정, 새로고침, 화면 전환과 같은 세부 상호작용을 포함하는 대표적인 동적 인터랙션 단위로 알려져 있다.¹²⁾ 이러한 요소들은 사용자가 시스템의 상태를 이해하고 다음 행동을 예측하도록 돕는 동시에, 제품 사용 과정에서의 몰입감과 사용성을 형성하는 데 중요한 역할을 수행한다.

그러나 동적 인터랙션은 정적인 UI 요소와 연속적인 피드백 과정을 포함하기 때문에 설계 의도를 명확하게 전달하기 어렵다.¹³⁾ 특히 타이밍(Timing), 지연(Delay), 피드백(Feedback), 전환(Transition)과 같은 요소들은 화면 설계서나 일반적인 디자인 시스템 문서

interactive digital technologies: A systematic literature review", HCI International 2020 - Late Breaking Papers: User Experience Design and Case Studies, Springer, 2020, pp.441-451.

9) Almaral Martinez, M. E., Berná-Martínez, J. V., Maciá Pérez, F., Lorenzo Fonseca, I., & Pascual Cáceres, C. P., "Understanding the importance of animation in human-computer interfaces and its relation to user experience", Proceedings of the 17th International Conference on Interfaces and Human Computer Interaction, ACM, 2023, pp.164-171.

10) Forlizzi, J., & Battarbee, K., "Understanding experience in interactive systems", Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '04), ACM, 2004, pp.261-268.

11) Norman, D. A., The Design of Everyday Things (Revised and Expanded Edition), Basic Books, 2013, pp.1-368.

12) Saffer, D., Microinteractions: Designing with details, O'Reilly Media, 2013, pp.1-160.

13) Leiva, G., Maudet, N., Mackay, W., & Beaudouin-Lafon, M., "Enact: Reducing designer-developer breakdowns when prototyping custom interactions", ACM Transactions on Computer-Human Interaction, 26(3), Article 19, 2019, pp.1-48.

만으로 충분히 표현되기 어렵다는 한계를 가진다.¹⁴⁾

동적 인터랙션은 단순히 전달되는 설계 산출물이 아니라 해석되고 구현되어야 하는 대상이라는 점에서, 협업 과정에서 발생하는 문제를 전달 방식이나 문서화 수준만으로 설명하기에는 한계가 있다.¹⁵⁾

최근 수행된 연구에서는 동적 인터랙션의 구현 및 전달 과정에서 디자이너와 개발자가 서로 다른 관점으로 인터랙션을 이해하며, 이러한 차이가 협업 과정의 갈등과 조율 부담으로 이어질 수 있음이 보고되었다.¹⁶⁾ 그러나 이러한 연구들은 주로 협업 과정에서 나타나는 현상과 경험을 중심으로 논의하였으며, 이러한 차이가 어떠한 인지적 구조에서 비롯되는지는 충분히 설명하지 못하였다.

특히 동일한 인터랙션 요소에 대해 디자이너는 사용자 가치(User Value)와 사용 맥락을 중심으로 이해하는 반면, 개발자는 개발 명세(Development Specification)와 시스템 동작을 중심으로 이해할 가능성이 있다.¹⁷⁾ 이러한 차이는 단순한 관점의 차이가 아니라 인터랙션을 구성하는 요소들을 분류하고 조직하는 방식의 차이로 연결될 수 있다.

따라서 동적 인터랙션 협업을 보다 깊이 이해하기 위해서는 협업 과정 자체를 넘어, 디자이너와 개발자가 동적 인터랙션을 어떠한 개념 체계로 이해하고 구조화하는지 탐색할 필요가 있다.

기존 연구들은 주로 디자인 핸드오프, 프로토타이핑, 디자인 시스템, 협업 도구와 같은 전달 매체 및 프로세스 개선에 초점을 두어 왔다. 또한 최근 연구들은 협업 과정에서 나타나는 해석 차이와 조율 경험을 탐색하고 있다. 그러나 디자이너와 개발자가 동적 인터랙션을 분

14) Maudet, N., Leiva, G., Beaudouin-Lafon, M., & Mackay, W. E., "Design breakdowns: Designer-developer gaps in representing and interpreting interactive systems", Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17), ACM, 2017, pp.630-641.

15) Leiva, G., Maudet, N., Mackay, W., & Beaudouin-Lafon, M., Op. cit., 2019, pp.1-48.

16) Hiix Lab, Papers, 2026. (2026.06.01.) <https://www.hiixlab.works/PAPERS>

17) Escamilla, A., Melenchón, J., Monzo, C., Morán, J. A., & Carrascal, J. P., "Exploring a user-centered approach for movement-based features in interaction design", International Journal of Human-Computer Studies, 2026, pp.1-22.

류하고 구조화하는 기준과 주요 요소에 대한 인식 차이를 탐색한 연구는 상대적으로 부족한 실정이다.

1-2. 연구 목적 및 질문

본 연구는 디자이너와 개발자가 동적 인터랙션을 어떠한 개념 체계로 이해하고 분류하는지 탐색하고, 직군별 인식 차이의 특성을 비교·분석하는 것을 목적으로 한다.

이를 위해 동적 인터랙션을 구성하는 요소를 사용자 가치(User Value), 개발 명세(Development Specification), 그리고 마이크로인터랙션 구성 요소(Trigger, Rules, Feedback, Loops and Modes)로 구분하고, 카드소팅(Card Sorting)과 개념도(Concept Mapping)를 활용하여 참여자들의 분류 방식, 개념화 방식, 중요도 인식을 조사한다.

본 연구의 연구 질문은 다음과 같다.

RQ1. 디자이너와 개발자는 동적 인터랙션의 구성 요소를 어떠한 기준으로 분류하고 구조화하는가?

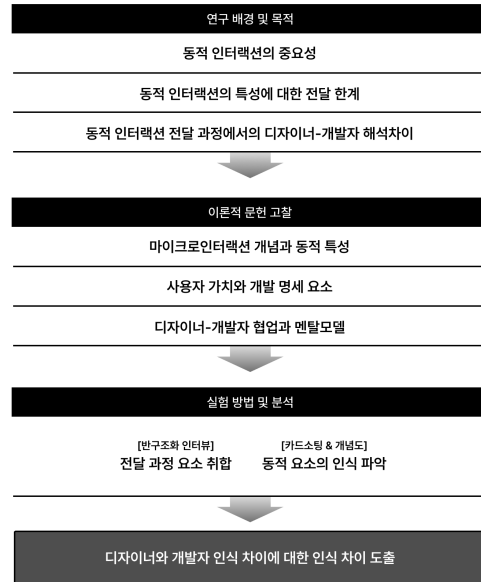
RQ2. 디자이너와 개발자는 동적 인터랙션 요소를 어떠한 개념으로 명명하고 설명하는가?

RQ3. 디자이너와 개발자는 동적 인터랙션 요소 중 무엇을 중요하게 인식하며, 그 이유는 무엇인가?

1-3. 연구 방법 및 범위

본 연구는 디지털 제품 환경에서 사용자 행동과 시스템 반응 사이에 발생하는 동적 인터랙션을 연구 대상으로 한다. 연구의 범위는 장식적 시각 효과가 아닌 기능적 역할을 수행하는 인터랙션 요소로 한정하며, 버튼 피드백, 상태 전환, 새로고침, 화면 전환과 같이 구현 과정에서 구체적인 전달과 해석이 요구되는 동적 UI 요소를 중심으로 설정하였다.

연구 방법은 이론적 문헌 연구, 반구조화 인터뷰, 카드소팅, 개념도 실험의 순서로 진행된다. 참여자는 동적 인터랙션 설계 및 구현 경험을 보유한 디지털 제품 디자이너 6인과 개발자 6인, 총 12인으로 구성하였다. 카드소팅을 통해 동적 인터랙션 구성 요소에 대한 직군별 분류 기준과 명명 방식을 분석하고, 중요도 선택 과정을 통해 직군별로 중요하게 인식하는 요소를 파악하였다.



[그림 1] 연구 전체 프로세스

2. 이론적 배경 및 연구 프레임워크

2-1. 동적 인터랙션과 마이크로인터랙션

동적 인터랙션(Dynamic Interaction)은 사용자의 행동과 시스템 상태 변화에 따라 시간의 흐름 속에서 전개되는 상호작용을 의미한다.¹⁸⁾ 정적인 인터페이스가 화면의 구성과 정보 표현에 초점을 둔다면, 동적 인터랙션은 행동, 반응, 전환, 상태 변화와 같은 시간 기반 경험을 포함한다.¹⁹⁾ 최근 디지털 제품의 사용자 경험은 단순한 화면 구성보다 사용자 행동에 대한 시스템의 반응과 경험의 흐름에 의해 차별화되는 경향을 보이며, 이에 따라 동적 인터랙션의 중요성이 더욱 강조되고 있다.²⁰⁾

마이크로인터랙션은 이러한 동적 인터랙션을 구성하는 대표적인 단위이다. Saffer(2013)는 마이크로인터랙션을 사용자가 특정 기능을 수행하기 위해 경험하는 작

18) Bétrancourt, M., & Boucheix, J.-M., Animation and motion in UX design. In Human-Computer Interaction, CRC Press, 2024, pp.1-24.

19) Arellano-Uson, J., Magaña, E., Morató, D., & Izal, M., "Survey on quality of experience evaluation for cloud-based interactive applications", Applied Sciences, 14(5), 2024, pp.1-35.

20) Ibid., pp.1-35.

은 규모의 상호작용으로 정의하고, 이를 트리거(Trigger), 동작 규칙(Rules), 피드백(Feedback), 순환과 모드(Loops and Modes)의 네 가지 요소로 설명하였다.²¹⁾ 마이크로인터랙션은 버튼 클릭, 호버, 로딩, 상태 전환과 같이 비교적 작은 상호작용으로 나타나지만, 사용자의 행동을 안내하고 시스템 상태를 전달하며 경험의 흐름을 형성하는 중요한 역할을 수행한다. 트리거는 인터랙션을 시작하는 발생 조건이며, 규칙은 인터랙션의 동작 규칙을 정의한다. 피드백은 사용자의 행동에 대한 시스템의 반응을 의미하며, 순환과 모드는 반복적 동작과 상태 변화를 조정하는 역할을 수행한다.

특히 마이크로인터랙션은 정적인 UI 요소와 달리 시간성과 연속성을 포함하는 동적 특성을 가진다. 사용자에게 제공되는 피드백은 단순한 화면 변화가 아니라 타이밍(Timing), 변화 곡선(Easing), 지연(Delay), 지속 시간(Duration)과 같은 시간적 속성의 조합을 통해 구현된다. Betz and Hall(2015)은 이러한 동적 인터페이스 속성이 정적 문서나 전통적인 설계 산출물만으로는 충분히 표현되기 어렵다고 지적하였다.²²⁾ 또한 김건동(2024)은 UI 피드백 모션의 타이밍과 변화 곡선을 정량적으로 분석하며, 마이크로인터랙션의 특성이 정성적 표현만으로는 충분히 기술되기 어려운 측면이 있음을 제시하였다.²³⁾ 이는 동적 인터랙션을 전달하고 구현하는 과정에서 설계 의도에 대한 해석 차이가 발생할 수 있음을 시사한다.

이러한 특성은 마이크로인터랙션이 단순한 시각 효과가 아니라 사용자 행동, 시스템 규칙, 상태 변화, 피드백, 반복 및 분기 구조가 결합된 복합적인 동적 요소임을 의미한다. 따라서 마이크로인터랙션의 설계와 구현은 설계 의도와 구현 조건을 공유하는 과정일 뿐만 아니라, 동적 속성의 의미와 경험적 가치를 함께 해석하는 과정으로 이해될 수 있다. 특히 동일한 인터랙션 요소라 하더라도 디자이너는 사용자 가치와 사용 맥락을 중심으로 이해하는 반면, 개발자는 구현 규칙과 시스템 동작을 중심으로 이해할 가능성이 있다.

그러나 기존 연구들은 주로 마이크로인터랙션의 설계 원칙, 사용자 경험 효과, 프로토타이핑 기법 또는 디자인 산출물의 전달 방식에 초점을 두어 왔다. 반면 디자이너와 개발자가 동적 인터랙션을 어떠한 개념 체계로 이해하고 구조화하는지, 그리고 직군별로 어떠한 요소를 중요하게 인식하는지에 대한 연구는 상대적으로 부족한 실정이다. 동적 인터랙션 협업 과정에서 발생하는 문제를 보다 깊이 이해하기 위해서는 전달 방식이나 협업 프로세스를 넘어, 참여자들이 인터랙션을 어떻게 개념화하고 조직하는지에 대한 멘탈모델 수준의 탐색이 필요하다.

2-2. 사용자 가치와 구현

디지털 제품 개발 과정은 사용자가 경험하게 될 가치를 설계하고, 이를 인터랙션 구조와 기술적 구현으로 구체화하는 과정으로 이해할 수 있다. 사용자 경험 관점에서 디자이너는 사용자가 인터페이스를 통해 어떠한 가치를 경험하는가에 주목하며, 개발자는 이러한 경험적 의도를 실제 시스템 동작으로 구현하기 위한 기술적 조건과 제약을 고려한다. 따라서 마이크로인터랙션은 단순한 인터랙션 구조를 넘어 사용자 가치와 개발 명세를 연결하는 매개체로 이해될 필요가 있다.

기존 마이크로인터랙션 연구는 주로 Saffer(2013)가 제안한 트리거, 동작 규칙, 피드백, 순환과 모드 네 가지의 구성요소를 중심으로 발전해 왔다.²⁴⁾ 이러한 구조는 사용자 행동과 시스템 반응 간의 관계를 설명하는 데 유용한 틀을 제공하였으며, 이후 다양한 연구에서 인터랙션 설계의 기본 프레임워크로 활용되었다.

그러나 최근 HCI 연구의 흐름은 인터랙션의 기능적 측면보다 인터랙션이 사용자에게 제공하는 경험적 가치에 더욱 주목하고 있다. Schrepp et al.(2026)의 연구에 의하면 HCI 연구가 전통적인 사용성(Usability) 중심 관점에서 감정(Emotion), 몰입(Engagement), 신뢰(Trust), 웰빙(Well-being) 등 사용자 경험 중심 관점으로 이동하고 있음을 보고하였다.²⁵⁾ Merz et al.(2016)은 모바일 인터페이스에서의 애니메이션 전환이 사용자 참여도, 감정적 반응, 인터페이스 이해도 향

21) Saffer, D., Op. cit., 2013, pp.1-160.

22) Betz, S., & Hall, R., "Self-archiving with ease in an institutional repository: Microinteractions and the user experience", Information Technology and Libraries, Vol. 34, No. 3, 2015, pp.44-58.

23) 김건동, "UI 컴포넌트 유형별 마이크로인터랙션의 트리거에 따른 피드백의 표현 분석", 디지털콘텐츠학회논문지, Vol. 25, No. 10, 2024, pp.2807-2818.

24) Saffer, D., Op. cit., 2013, pp.1-160.

25) Schrepp, M., Rosenzweig, E., Soares, M. M., & Marcus, A., "A review of design, user experience, and usability in HCI, through the lens of the DUXU conference", International Journal of Human-Computer Interaction, 42(3), 2025, pp.1284-1309.

상에 기여함을 실증적으로 확인하였다.²⁶⁾ Kolte and Rao(2024)는 마이크로인터랙션이 단순히 사용성을 향상시키는 기능적 요소를 넘어, 인터페이스의 개성(Personality), 즉각적인 반응성(Responsiveness), 그리고 즐거움(Delight)을 부여함으로써 보다 풍부한 사용자 경험을 형성하는데 기여한다고 주장하였다.²⁷⁾ 이러한 연구들은 마이크로인터랙션이 단순한 인터랙션 구조가 아니라 사용자 경험 형성을 통한 사용자 가치 실현에 중요한 역할을 수행함을 보여준다.

한편 사용자 가치가 실제 제품으로 구현되기 위해서는 개발 가능한 형태의 명세(Development Specification)로 구체화되어야 한다. 그러나 사용자 경험과 구현 사이의 과정을 체계적으로 설명한 연구는 상대적으로 제한적이다. Almaral Martinez et al.(2023)은 인터페이스 애니메이션 설계가 여전히 경험과 직관에 의존하는 경우가 많으며, 체계적인 설계 프레임워크가 부족하다고 지적하였다.²⁸⁾ Escamilla et al.(2026)은 움직임 기반 인터랙션 연구에서 기술적 파라미터(Velocity, Acceleration, Joint Angle 등)를 디자이너가 이해할 수 있는 의미 중심 특징(Designer-interpretable Features)으로 변환할 필요성을 강조하였다.²⁹⁾ 또한 Sousa et al.(2019)은 사용자 인터랙션 명세를 UX 의도가 구현 가능한 규칙으로 변환되는 과정을 설명하였다.³⁰⁾ 이는 사용자 가치가 실제 시스템 동작으로 구현되기 위해서는 명시적인 인터랙션 명세가 필요함을 보여준다.

2-3. 디자이너-개발자 협업 과정의 멘탈모델의 중요성

멘탈모델(Mental Model)은 개인이 특정 시스템이나 현상의 구조와 작동 원리를 이해하고 예측하기 위해 형성하는 내적 표상(Internal Representation)을 의미한다. Norman(2013)은 사용자가 시스템의 동작을 이해하고 예측하기 위해 형성하는 개념적 모형으로 설명하였다.³¹⁾ HCI 분야에서 멘탈모델은 사용자가 시스템을 어떻게 이해하고 사용하는지를 설명하는 핵심 개념으로 활용되어 왔으며, Rogers et al.(2007)은 효과적인 인터랙션 설계를 위해 사용자의 멘탈모델과 시스템 모델 간의 정합성이 중요하다고 설명하였다.³²⁾ 이러한 연구들은 동일한 시스템이라 하더라도 개인이 형성하는 멘탈모델에 따라 이해와 행동이 달라질 수 있음을 보여준다.

협업 환경에서도 멘탈모델은 중요한 역할을 수행한다. Van der Veer et al.(1996)은 동일한 시스템에 대해서도 사용자, 디자이너, 개발자가 서로 다른 멘탈모델을 형성할 수 있으며, 이러한 차이가 시스템 이해와 협업 과정에서의 문제를 유발할 수 있다고 주장하였다.³³⁾ 즉 협업 과정에서 공유되는 산출물이나 정보가 동일하더라도 각 참여자는 자신의 전문 지식과 경험을 기반으로 이를 다르게 해석할 수 있으며, 이러한 해석의 차이는 의사결정과 협업 결과에 영향을 미칠 수 있다. 따라서 협업에서는 단순히 정보가 전달되었는 가보다, 참여자들이 해당 정보를 얼마나 유사한 방식으로 이해하고 있는지가 중요한 요인으로 작용할 수 있다.

실제로 디자이너와 개발자 간 협업 연구에서도 이러한 문제는 반복적으로 보고되고 있다. Maudet et al.(2017)은 설계 전달 과정에서 협업 단절(Design Breakdown)이 발생하는 주요 원인 중 하나로 디자인 산출물에 대한 상이한 해석을 지적하였다.³⁴⁾ 또한 Feng et al.(2023)은 디자인과 개발 간 정렬(Alignment)을 유지하기 위한 문서화와 주석 체계의

26) Merz, B., Tuch, A. N., & Opwis, K., "Perceived user experience of animated transitions in mobile user interfaces", *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2016, pp.3152-3158.

27) Kolte, A., & Rao, D., "Exploring Microinteractions in Human-Computer Interaction: Design Principles, Types, and User Experience", *International Conference on Human-Centric Smart Computing*, Springer Nature Singapore, 2024, pp.13-23.

28) Almaral Martinez, M. E., Berna Martinez, J. V., Macia Perez, F., Lorenzo Fonseca, I., & Pascual Caceres, C. P., *Op. cit.*, 2023, pp.164-171.

29) Escamilla, A., Melenchón, J., Monzo, C., Morán, J. A., & Carrascal, J. P., *Op. cit.*, 2026, pp.1-22.

30) Sousa, V., Syriani, E., & Fall, K., "Operationalizing the integration of user interaction specifications in the synthesis of modeling editors", *Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering*, 2019, pp.42-54.

31) Norman, D. A., *Op. cit.*, 2013, pp.1-368.

32) Sharp, H., Jenny, P., & Rogers, Y., *Interaction Design: Beyond Human-Computer Interaction*, Aufl., Chichester, 2007, pp.1-518

33) Van Der Veer, G. C., Lenting, B. F., & Bergevoet, B. A., "GTA: Groupware task analysis-Modeling complexity", *Acta psychologica*, 91(3), 1996, p.297.

34) Maudet, N., Leiva, G., Beaudouin-Lafon, M., & Mackay, W., *Op. cit.*, 2017, pp.630-641.

중요성을 강조하였으며,³⁵⁾ Zhang et al.(2025)은 디자이너와 개발자가 동일한 아티팩트를 서로 다른 관점에서 해석함으로써 상호 이해 부족이 발생할 수 있음을 보고하였다.³⁶⁾ 특히 디자이너는 사용자 경험과 인터랙션 의도를 중심으로 설계를 이해하는 반면, 개발자는 이를 구현 가능성과 기술적 제약의 관점에서 해석하는 경향이 있다. 이는 협업 과정에서 발생하는 문제의 원인이 단순한 커뮤니케이션 부족이 아니라, 설계 대상을 바라보는 인지적 틀 자체의 차이에 있을 가능성을 시사한다.

이러한 관점은 마이크로인터랙션과 같은 동적 인터랙션에서 더욱 중요하다. 동적 인터랙션은 사용자 가치, 행동 규칙, 피드백, 순환과 모드, 그리고 개발 명세가 복합적으로 연결된 구조를 가지므로 정적인 인터페이스 요소보다 해석의 여지가 크다. 따라서 디자이너와 개발자가 동적 인터랙션을 어떠한 개념 구조로 이해하고, 어떤 요소를 중요하게 인식하는지 확인하는 것은 협업 과정에서 발생하는 인식 차이의 근본 원인을 이해하는 데 중요하다. 나아가 이러한 멘탈모델의 차이를 파악하는 것은 효과적인 협업 체계 구축을 위한 기반을 제공할 수 있다.

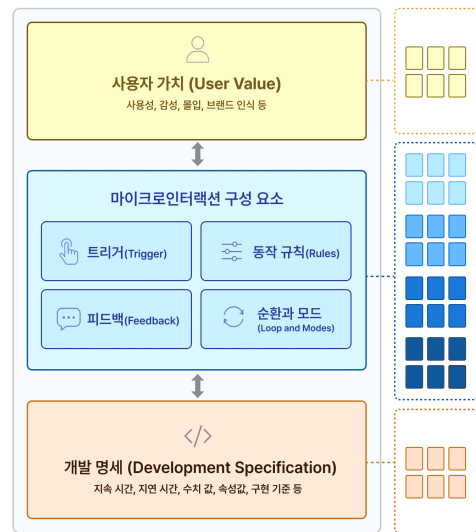
2-4. 분석을 위한 개념적 프레임워크

본 연구는 동적 인터랙션을 사용자 가치, 개발 명세, 그리고 마이크로인터랙션의 구성 요소 간 관계를 중심으로 이해한다. 구체적으로 마이크로인터랙션의 구성 요소인 트리거, 동작 규칙, 피드백, 순환과 모드를 포함하여 동적 인터랙션을 통합적으로 분석한다. 이러한 관점은 동적 인터랙션 관련 선행연구에서 제시된 개념들을 본 연구의 목적에 맞게 재구성한 것이다.

이를 기반으로 본 연구는 카드소팅과 개념도 실험을 수행하여, 디자이너와 개발자가 동적 인터랙션 구성 요소를 어떻게 분류하는지 조사하였다. 구체적으로는 첫째, 사용자 가치, 개발 명세, 마이크로인터랙션 구성 요

소가 직군별로 어떻게 분류되고 구조화되는지 확인하였다. 둘째, 동적 인터랙션 요소가 어떤 기준으로 명명되고, 어떤 요소에 중요도가 부여되는지 분석하였다. 셋째, 이러한 인식 구조 차이가 협업 과정에서 어떤 문제를 유발할 수 있으며, 어떤 시사점을 제공하는지 논의하였다.

본 연구는 이러한 동적 인터랙션에 대한 디자이너와 개발자의 인식 구조의 특성과 차이를 해석하고자 한다. 이를 통해 동적 인터랙션의 설계 및 구현 과정에서 발생하는 직군 간의 해석 차이를 줄이고, 보다 원활한 협업 환경을 조성하는 데 기여할 수 있기를 기대한다.



[그림 2] 개념적 프레임워크

3. 연구 방법

3-1. 연구 설계

본 연구는 디자이너와 개발자가 동적 인터랙션을 어떻게 이해하고 구조화하는지 탐색하고, 직군 간 인식 차이를 분석하기 위해 카드소팅과 개념도를 활용한 탐색 연구를 수행하였다. 카드소팅은 참여자가 개념 간 유사성과 관계성을 바탕으로 정보를 분류하는 과정을 통해 인지 구조와 개념 체계를 탐색하는 연구 방법이다. Rugg et al.(2022)는 카드소팅을 사용자의 개념 조직 방식을 파악하기 위한 대표적인 기법으로 설명하였으며, 특히 참여자가 직접 범주를 구성하고 명명하는

35) Feng, K. J. K., Li, T. W., & Zhang, A. X., "Understanding collaborative practices and tools of professional UX practitioners in software organizations", Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023, pp.1-20.

36) Zhang, S., Zhang, T., Cheng, J., & Zhou, S., "Who is to blame: A comprehensive review of challenges and opportunities in designer-developer collaboration", Proceedings of the ACM on Human-Computer Interaction, 9(2), 2025, pp.1-32.

개방형(Open Card Sorting) 방식은 새로운 분류 체계를 탐색하는데 적합하다고 제안하였다.³⁷⁾ 또한 Tchivi et al.(2025)은 50편의 관련 연구를 검토한 결과, 카드소팅이 정보 구조와 개념 분류 체계를 탐색하는데 효과적인 방법임을 보고하였다.³⁸⁾ 이에 본 연구는 참여자가 동적 인터랙션 요소를 어떠한 기준으로 분류하고 이해하는지 확인하기 위해 개방형 카드소팅 방식을 적용하였다. 이후 개념도를 통해 카드소팅에서 분류된 요소들이 참여자의 인식 속에서 어떻게 연결되는지 확인하였다.

3-2. 연구 대상 및 참여자

연구 참여자는 총 12명으로 구성되었다. 디자이너 그룹 6명과 개발자 그룹 6명을 모집하였으며, 마이크로인터랙션을 포함한 동적 인터랙션 관련 설계와 협업 경험을 보유한 자로 선정하였다. 참여자 선정 기준은 다음과 같다.

동적 인터랙션이 포함된 서비스 프로덕트에 대한 설계와 협업 경험 보유

디자이너 그룹 : 경력 3년 이상의 개발자와 협업 경험이 있는 UI/UX 디자이너 또는 프로덕트 디자이너

개발자 그룹 : 경력 3년 이상의 디자이너와의 협업 경험이 있는 프론트엔드 또는 풀스택 개발자

참여자는 채용, 교육, 여행, AI 서비스, 핀테크 등 다양한 산업 도메인과 스타트업부터 대기업에 이르는 다양한 조직 규모에 분포하도록 구성하여, 폭넓은 협업 맥락이 반영될 수 있도록 하였다.

분석 과정에서는 익명성을 보장하고 역할 구분을 명확히 하기 위하여 디자이너 그룹은 P1-D~P1-6D, 개발자 그룹은 P7-F~P12-F로 코드화하여 표기하였다. 참여자별 세부 정보는 [표 1]과 같다.

[표 1] 연구 참여자 정보

참여자	직무	실무 경력	소속 규모	도메인
디자이너				

37) Rugg, G., & McGeorge, P., "The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts", Expert Systems, 14(2), 2002, pp.80-93.

38) Tchivi, E., Sharma, B., & Paea, S., "A systematic review of the comparison of different types of card sorting", IEEE Access, 2025, pp.52334-52352.

P1-D	프로덕트 디자이너	3-5년	50-300명	채용 플랫폼
P2-D	프로덕트 디자이너	6-10년	50명 미만	에이전시
P3-D	프로덕트 디자이너	3-5년	50명 미만	교육 플랫폼
P4-D	프로덕트 디자이너	6-10년	50명 미만	여행 플랫폼
P5-D	UI/UX 디자이너	3-5년	50-300명	팬덤 플랫폼
P6-D	프로덕트 디자이너	3-5년	프리랜서	디지털 제품
개발자				
P7-F	프론트엔드 개발자	6-10년	50-300명	소프트웨어
P8-F	프론트엔드 개발자	6-10년	1,000 -10,000명	배달 플랫폼
P9-F	프론트엔드 개발자	3-5년	50명 미만	AI 서비스
P10-F	프론트엔드 개발자	3-5년	1,000 -10,000명	핀테크
P11-F	풀스택 개발자	6-10년	50 - 300명	패션 플랫폼
P12-F	풀스택 개발자	10년 이상	50명 미만	영업 지원 플랫폼

3-3. 카드범주의 도출 및 구성

카드 세트는 동적 인터랙션에 대한 이론적 관점과 실무적 관점을 통합하여 총 36개의 카드로 구성하였다. 카드 범주는 사용자 가치 범주, 마이크로인터랙션의 구성 요소 범주, 개발 명세 범주의 세 영역으로 구성되었다.

첫째, 사용자 가치 범주는 동적 인터랙션이 사용자 경험에 미치는 가치적 측면을 반영하였다. 관련 연구에서는 사용성뿐만 아니라 감성적 만족과 브랜드 경험이 중요한 UX 요소로 제시되어 왔다.³⁹⁾⁴⁰⁾ 이에 자연스러움, 부드러움, 예측 가능성, 몰입감, 브랜드 감성, 사용성을 포함하였다.

둘째, 마이크로인터랙션 구성 요소는 Saffer(2013)의 프레임워크를 기반으로 트리거, 동작 규칙, 피드백, 순환과 모드의 네 범주로 구성하였다. 본 연구에서는

39) Hassenzahl, M., Hedonic, emotional, and experiential perspectives on product quality. In Encyclopedia of human computer interaction, IGI Global Scientific Publishing, 2006, pp. 266-272.

40) Aagesen, P. T., & Heyer, C., &Personality of interaction: Expressing brand personalities through interaction aesthetics&, In Proceedings of the 2016 CHI conference on human factors in computing systems, 2016, pp.3126-3130.

루프와 모드가 시간과 조건에 따른 인터랙션이 반복, 지속, 전환, 종료되는 구조와 관련된다는 점을 바탕으로 상태(State) 개념을 함께 포함하였다.

셋째, 개발 명세 범주는 동적 인터랙션 구현에 사용되는 주요 기술적 속성을 중심으로 구성하였다. 이에 따라 상태 전환의 속도를 결정하는 전환 속도(Transition Speed), 애니메이션의 가속감속 패턴을 정의하는 변화 곡선(Easing), 물리 기반 움직임의 특성을 제어하는 스프링 값(Spring Value), 요소의 시각적 가시성을 조절하는 불투명도(Opacity), 요소의 확대축소를 나타내는 크기 변화(Scale), 그리고 시스템의 처리 상태를 반영하는 API 상태(API State)를 세부 요소로 포함하였다. 카드 구성은 [표 2]에 제시했다.

[표 2] 카드 범주 구성

범주	코드	컬러	카드 구성
사용자 가치 (User Value)	UV	옐로우 (#FFD600)	자연스러운, 부드러운, 예측 가능성, 몰입감, 브랜드 감성, 사용성
트리거 (Trigger)	MT	라이트 블루 (#BFE8FF)	탭(Tap), 드래그(Drag), 스와이프(Swipe), 롱프레스(Long Press), 당기기(Pull), 스크롤(Scroll)
동작 규칙 (Rules)	MR	스카이 블루 (#62B5F6)	지속 시간(Duration), 지연(Delay), 임계점(Threshold), 반복 조건, 예외 처리, 상태 분기
피드백 (Feedback)	MF	블루 (#1976D2)	애니메이션(Animation), 진동(Haptic), 색상 변화, 사운드(Sound), 토스트 메시지(Toast Message), 로딩 인디케이터>Loading Indicator)
상태와 모드 (State/Loops & Modes)	ML	다크 블루 (#0B4F8A)	기본 상태(Default), 로딩 상태>Loading), 완료 상태>Success), 실패 상태>Error), 비활성화 상태>Disabled), 전환 상태>Transition)
개발 명세 (Development Specification)	DS	오렌지 (#FF9800)	Easing(변화 곡선), Spring Value (스프링 값), Opacity(불투명도), Scale(크기 변화), Transition Speed

		(전환 속도), API 상태
--	--	-----------------

3-4. 실험 절차

실험은 온라인 화상 도구인 Zoom을 활용한 1:1 온라인 방식으로 진행되었으며, 카드소팅과 개념도 작업은 온라인 협업 도구인 Miro(Miro.com)를 활용하여 수행하였다. 실험 절차는 설명 및 참여 동의, 반구조화 인터뷰, 카드소팅, 카드소팅 사후 인터뷰, 개념도 작성, 실험 종료 안내 순서로 진행하였다.

반구조화 인터뷰에서는 동적 인터랙션의 정의, 협업 경험, 전달 방식, 중요 정보, 마이크로인터랙션 구성 요소에 대한 이해를 조사하였다. 인터뷰 항목은 [표 4]에 제시하였다.

개념도 단계에서는 카드소팅 과정에서 형성된 그룹을 기반으로, 요소 간 관계와 흐름 구조를 자유롭게 표현하도록 하였다. 이후 작성된 개념도를 바탕으로 참여자의 요소 간 연결 방식과 인식 흐름을 확인하였다.

[표 3] 실험 절차

단계	절차	수행 내용
1	연구 설명 및 동의	연구 목적 설명 및 실험 동의 확인
2	실험 절차 안내	카드소팅 및 개념도 수행 방법 안내
3	반구조화 인터뷰	동적 인터랙션의 정의, 협업 경험, 마이크로인터랙션 구성 요소 4가지 인식
4	카드소팅 수행	36개 카드를 자유롭게 분류하고 그룹명 작성
5	카드소팅 사후 인터뷰	분류 기준, 중요 그룹, 모호한 요소 탐색
6	개념도 구성	카드 그룹간 관계 및 흐름 연결
7	실험 종료 안내	추가 의견 수집 및 연구 종료 안내

[표 4] 반구조화 인터뷰 질문 항목

항목	No.	질문
동적 인터랙션의 정의	1	동적 인터랙션을 어떻게 정의하시나요?
협업 및 전달 방식	2	디자이너/개발자에게 동적 인터랙션을 설명할 때 주로 사용하는 도구와 전달 방식은 무엇인가요?
핵심 전달	3	동적 인터랙션 정보 중 협업 과정에서

정보		반드시 전달되어야 하는 핵심 요소는 무엇인가요?
협업 과정의 어려움	4	동적 인터랙션을 설명하거나 전달하는 과정에서 가장 어렵다고 느끼는 부분은 무엇인가요?
핵심 요소에 대한 실무적 의견	5	마이크로인터랙션 주요 구성 요소 트리거&규칙&피드백&루프와 모드의 네 가지 개념을 알고 계시나요?

3-5. 분석 방법

자료 분석은 카드소팅 결과와 개념도 결과를 중심으로 수행하였다.

첫째, 카드소팅에서 생성된 카드 구성과 그룹명을 분석하여 참여자들이 동적 인터랙션 요소를 어떠한 개념으로 해석하는지 확인하였다.

둘째, 참여자가 중요하다고 판단한 그룹을 분석하여 동적 인터랙션에서 우선적으로 고려되는 요소를 확인하고 직군간 차이를 확인하였다.

셋째, 카드 그룹 범주별 응집도를 분석하여 어떤 요소가 독립적으로 인식되고, 어떤 요소가 다른 범주와 함께 연결되어 해석되는지 확인하였다.

마지막으로 실험 결과를 해석하여, 디자이너와 개발자가 형성하고 있는 동적 인터랙션 구성 요소를 분류하고 구조화하는 방식의 공통점과 차이점을 도출하였다.

4. 연구 결과

4-1. 카드 그룹화 결과

카드소팅 분석 결과, 본 연구에 참여한 디자이너와 개발자 모두 마이크로인터랙션의 학술적 구성 요소를 명시적으로 학습하지 않았음에도 불구하고, 실험 과정에서 제시된 6개의 범주를 전반적으로 구분하여 인식하는 경향을 보였다. 그러나 각 요소를 해석하고 구조화하는 방식에서는 직군별로 다른 패턴이 관찰되었다.

본 연구에서는 카드소팅 결과로부터 도출된 유사도 행렬을 기반으로 네트워크 및 군집 분석을 수행하는 기존 연구의 방법을 참고하여, 범주 내 카드들 간 평균 연결 강도를 응집도(Cohesion)의 척도로 활용하였다.⁴¹⁾ 응집도는 동일 범주에 속한 카드들이 참여자들

에 의해 함께 분류된 정도를 의미한다. 본 연구에서는 각 범주 내 카드쌍이 함께 묶인 횟수의 합을 가능한 카드쌍 수와 참여자 수의 곱으로 나누어 응집도를 계산하였다.

응집도 = 두 카드가 함께 묶인 횟수의 합 / (가능한 카드쌍 수 × 참여자 수)

응집도가 높을수록 해당 범주의 카드들이 참여자들에 의해 함께 분류되는 빈도가 높음을 나타내며, 낮을수록 해당 카드들이 다른 범주와 혼재되어 분류되고 있음을 나타낸다. 본 연구에서는 응집도를 직군 간 통계적 차이를 검증하기 위한 지표가 아닌, 카드소팅 결과의 분류 일관성과 참여자들의 분류 경향을 살펴보기 위한 기술적(Descriptive) 지표로 활용하였다. 본 연구에서는 표본 규모의 한계로 인해 직군 간 응집도 차이에 대한 통계적 유의성 검증은 수행하지 않았으며, 두 직군 간의 패턴에 대한 통계적 유의 여부에 대해서는 후속 연구를 통한 검증이 필요하다.



[그림 3] 높은 응집도를 보이는 범주의 카드소팅 결과 : 사용자 가치, 트리거, 상태와 모드

41) 이동일, 이혜준, "소비자 집단 인터뷰에서 의미



[그림 4] 상대적으로 낮은 응집도를 보이는 범주의 카드소팅 결과 : 동작 규칙, 피드백, 개발 명세

분석 결과, 가장 높은 응집도를 보인 범주는 ‘트리거’와 ‘상태와 모드’였다. 트리거는 디자이너와 개발자 모두 92%의 응집도를 보였으며, 상태와 모드는 디자이너 90%, 개발자 94%의 응집도를 나타냈다. 상태와 모드 범주는 개발자 직군에서 대부분 하나의 그룹으로 분류되었으며, 디자이너 직군에서는 개발 명세 요소에 해당하는 API 상태 카드까지 동일 범주에 포함하는 사례가 일부 관찰되었다. 이는 사용자 입력 행위와 시스템 상태 변화가 두 직군 모두에게 비교적 안정적이고 일관된 개념 범주로 인식되고 있음을 보여준다.

사용자 가치(User Value) 또한 높은 응집도를 나타냈다. 개발자 집단에서도 사용자 가치 요소가 별도의 범주로 형성되었다는 점은, 개발자 역시 동적 인터랙션의 경험적 가치를 비교적 명확하게 인식하고 있음을 보여준다.

동작 규칙(Rules) 범주에서는 두 직군 간 차이가 상대적으로 뚜렷하게 나타났다. 디자이너는 동작 규칙 범주의 카드를 하나의 동질적인 집단으로 묶거나 일부 다른 범주의 카드와 함께 묶는 경향을 보인 반면, 개발자는 개발 명세 요소에 해당하는 API 상태 카드까지 포함하여 구조화하였다.

피드백(Feedback) 범주에서도 두 직군 간 차이가 상대적으로 분명하게 나타났다. 디자이너는 피드백 범주의 카드를 하나의 동질적인 집단으로 묶거나 다른 범주의 카드와 함께 분류하는 경향을 보였으며, 개발자는 피드백 요소를 하나의 동질적인 집단으로 배치하는 경향을 보였다. 카드소팅 결과에서 나타난 빈칸은 참여자들이 해당 요소를 동일 범주의 구성 요소로 인식하지 않고 다른 범주와 함께 분류하였음을 의미한다. 이는 피드백 범주가 상태 변화, 동작 규칙, 개발 명세 등 인접 범주와 혼합되어 해석될 가능성이 있음을 보여준다.

개발 명세(Development Specification) 범주에서는 디자이너와 개발자 모두 동작 규칙이나 피드백 범주의 카드와 함께 묶는 경향이 나타났다. 이러한 결과는 [표 5]에 제시된 결과를 통해 확인할 수 있다.

[표 5] 동적 인터랙션 범주별 응집도 결과

범주별 응집도			
범주	디자이너	개발자	특징
사용자 가치	89%	83%	공통 인식
트리거	92%	92%	공통 인식
동작 규칙	83%	80%	State/DS와 혼합
피드백	67%	67%	가장 모호
상태와 모드	90%	94%	공통 인식
개발 명세	83%	92%	직군 차이 큼

4.2. 카드 그룹 명명 결과

카드소팅 그룹 명명 결과, 참여자들이 구성한 그룹은 전반적으로 연구자의 상위 범주와 일정한 대응 관계를 보였다. 다만 각 그룹이 연구자 범주와 일대일로 완전히 대응하는 것은 아니므로, 본 연구에서는 각 그룹에서 우세하게 나타난 카드의 성격과 참여자가 중심적으로 배치한 요소를 기준으로 인식 방향을 해석하였다.

각 그룹에 부여된 명칭을 분석한 결과, 디자이너와 개발자는 동일한 카드 집합에 대해서도 서로 다른 해석 체계를 적용하고 있는 것으로 나타났다.

사용자 가치 범주에서 디자이너는 ‘브랜드’, ‘사용자

경험', 'UX 원칙과 같은 표현을 사용하였다. 반면 개발자는 'UX', '디자인 언어', '사용자 경험에서의 판단 기준', '추상 개념' 등으로 명명하였다. 이는 디자이너가 사용자 가치를 사용자가 경험해야 할 목표나 방향성으로 인식하는 반면, 개발자는 설계 의도를 이해하기 위한 개념적 기준으로 해석하고 있음을 보여준다.

트리거 범주에서 디자이너는 '제스처', '액션', '유저 액션' 등 사용자의 행동을 중심으로 명명한 반면, 개발자는 'input', 'Event', '인터랙션 타입' 등 시스템에 전달되는 입력의 관점에서 명명하였다.

동작 규칙의 경우 디자이너는 '조건', '기획', '상태 인지'와 같은 표현을 사용하였고, 개발자는 '비즈니스 로직', 'Flow Chart', '개발자 언어와 같은 표현을 사용하였다. 이는 디자이너가 규칙을 사용자 경험의 흐름을 구성하는 조건으로 이해하는 반면, 개발자는 시스템 동작을 제어하기 위한 논리 구조로 이해하고 있음을 보여준다.

피드백 범주에서는 디자이너가 '피드백', '인터랙션', '애니메이션', '사용성' 등의 명명을 사용하였다. 이는 피드백을 사용자가 시스템 상태를 인지하고 경험의 질을 판단하는 경험적 요소로 이해하고 있음을 보여준다. 반면 개발자는 '상호작용(Output)', '시각적 인터랙션(UI)', '액션 피드백' 등의 명명을 사용하였다. 이는 피드백을 사용자 경험의 관점보다 시스템 동작의 결과를 전달하는 구현 요소로 해석하고 있음을 나타낸다.

상태와 모드 범주에서는 '상태', '상태값', '상태 및 피드백', '상태 인지'와 같은 명명이 나타났다. 디자이너는 로딩 상태, 전환 상태 등을 사용자가 화면에서 인지하는 상황과 변화 단계로 이해한다고 볼 수 있다. 개발자는 '상태주기변화', 'UI의 상태', 'State' 등의 표현을 사용하였다. 이는 상태와 모드를 시스템의 내부 상태와 상태 전이 로직을 설명하는 개념으로 인식하고 있음을 나타낸다.

개발 명세 범주에서 디자이너는 '효과', '애니메이션', '디자인 값(수치)' 등의 명명을 사용하였다. 디자이너는 개발 명세를 구현 정보 자체보다 움직임의 속도, 질감, 표현 효과를 조정하는 설계 기준으로 인식하였다. 반면 개발자는 '모션 값/모션 명세', '애니메이션을 결정짓는 수치', 'Design Token' 등의 응답을 하였다. 이는 개발자가 개발 명세를 인터랙션의 동작과 시각적 결과를 구현하기 위한 기술적 속성으로 인식하고 있음을 보여준다.

이러한 결과는 동일한 구성 요소를 다루고 있음에도

불구하고, 각 직군이 이를 서로 다른 개념 체계 안에서 해석하고 있음을 시사한다. 명명 결과의 시각화는 [그림 5]에 제시하였다.

디자이너 카드 그룹 명명	상위 범주	개발자 카드 그룹 명명
브랜딩 UX 원칙 평가 지표 사용자 경험 브랜딩 브랜딩 스타일 애니메이션	사용자 가치 (User Value)	UX 디자이너의 말 디자인 언어 사용자 경험에서의 판단 기준 추상 개념 UX
제스처 액션 모션 사용성 제스처 유저 액션		마이크로 인터랙션 상호작용(input) 인터랙션 타입 물리적 인터랙션 Event 유저 액션
조건, 스펙 로직 조건 기획 기획 상태 인지		unmounted 개발자들의 말 개발자 언어 UI에서 상태 전이 고려조건 Flow Chart 비즈니스 로직, 알고리즘
피드백 인터랙션, GUI 컴포넌트 요소 - 애니메이션 피드백		- 상호작용(Output) 디자인 시스템 시각적 인터랙션 Component 액션 피드백
상태 상태 상태 상태값 상태 및 피드백 실패 설명 제공		상태주기변화 컴포넌트 상태(Prop) State UI의 상태 State 애니메이션, uiux 상태
스펙 속성 효과 애니메이션 디자인 수치값 탈색	개발 명세 (Development Specification)	UX / UI의 기본변화 모션 값, 모션 명세 Property 애니메이션을 결정짓는 수치 Design Token 애니메이션 속성

[그림 5] 디자이너와 개발자 간 카드 그룹 요소 분포

4-3. 중요도 인식 비교

카드소팅 후 사후 인터뷰에서는 참여자가 구성한 그룹 중 가장 중요하다고 판단한 그룹을 선택하도록 하였다. 이후 해당 중요 그룹에 포함된 카드와 그룹명을 재분석한 결과, 두 직군 모두 '상태와 모드' 범주에 해당하는 상태 요소를 가장 중요하게 인식한 것으로 나타났다.

디자이너의 중요 그룹에 포함된 요소는 주로 '상태와 모드', '트리거', '피드백' 범주에 분포하였다. 이는 디자이너가 기본 상태, 전환 상태, 로딩 상태, 성공 상태, 실패 상태, 비활성화 상태와 같은 상태 관련 카드를 사용자가 화면에서 인지하는 상황과 변화 단계로 이해했음을 시사한다. 또한 트리거는 상태 변화가 시작되는 사용자 행동으로, 피드백은 변화된 상태를 사용자가 인지하도록 돕는 반응 요소로 함께 고려되었다.

반면 개발자의 중요 그룹에 포함된 요소는 '상태와 모드', '개발 명세', '동작 규칙' 범주에 분포하였다. 개발자는 상태 변화를 UI 상태값이자 컴포넌트가 관리해야 하는 상태 주기로 이해하였다. 또한 '지속 시간', '자연 시간', '임계값과 같은 개발 명세 요소와 동작 규칙 요소를 함께 고려하였다. 이는 개발자가 상태 변화를 조건에 따라 전환되며, 수치와 속성값을 통해 구현되는 시스템 구조로 인식하고 있음을 보여준다. 해당 내용은 [그림 6]에 제시하였다.

연구자 상위 범주	P1-D	P2-D	P3-D	P4-D	P5-D	P6-D
사용자 가치 (User Value)	브랜드, UX원칙	평가지표	사용자 경험	브랜드성	브랜드 스타일	애니메이션
트리거 (Trigger)	제스처 [아이콘]	액션	모션 [아이콘]	사용성 [아이콘]	제스처	유저 액션
동작 규칙 (Rules)	조건, 스펙 [아이콘]	로직 [아이콘]	조건	기획	기획	상태 인지 [아이콘]
피드백 (Feedback)	피드백	인터랙션, GUI 합목선도	요소	(일부)패널-레이어, 사용성	애니메이션	피드백
상태와 모드 (State, Loops and Modes)	상태 [아이콘]	상태 [아이콘]	상태 [아이콘]	상태값 [아이콘]	상태 및 피드백 [아이콘]	상태 설명 제공 [아이콘]
개발 명세 (Development Specification)	스펙	속성	효과	애니메이션	디자인 수치값	탐색

연구자 상위 범주	P7-F	P8-F	P9-F	P10-F	P11-F	P12-F
사용자 가치 (User Value)	UX	디자인의 열	디자인 언어	사용성/사용자 경험에서의 판단 기준	추상 개념	UX
트리거 (Trigger)	아이프로 인터랙션	상호작용 (Input)	인터랙션 타입	물리적 인터랙션	Event [아이콘]	유저 액션
동작 규칙 (Rules)	unmounted [아이콘]	개발자들의 말	개발자 언어	UI에서 상태 전이 고려조건	Flow Chart	비즈니스 로직, 알고리즘 [아이콘]
피드백 (Feedback)	(일부)UX / UI의 기본변화	상호작용 (Output)	디자인 시스템, 인터랙션 타입 일부	시각적 인터랙션	Component [아이콘]	액션 피드백
상태와 모드 (State, Loops and Modes)	상태주기변화	컴포넌트 상태(Prop)	State [아이콘]	UI의 상태 [아이콘]	State [아이콘]	애니메이션, uxui 상태
개발 명세 (Development Specification)	UX / UI의 기본변화	모션 리모션 명세 [아이콘]	Property	애니메이션을 결정짓는 수치 [아이콘]	Design Token [아이콘]	애니메이션 속성

[그림 6] 디자이너와 개발자 간 중요 인식 그룹

4-4. 직군 간 인식 차이 종합

실험 결과를 종합하면, 본 연구에 참여한 디자이너와 개발자는 동일한 동적 인터랙션 구성 요소를 다루고 있음에도 이를 인식하고 분류하며 다른 요소와 연결하는 방식에서 차이를 보이는 경향이 관찰되었다.

디자이너 참여자들은 사용자 가치, 트리거, 피드백, 상태와 모드가 연결되는 경험 중심의 구조를 형성한 반면, 개발자 참여자들은 입력, 규칙, 상태, 속성값, 출력 결과가 연결되는 구현 중심의 구조가 관찰되었다.

주목할 점은 참여자들이 Saffer(2013)의 마이크로

인터랙션 이론을 명시적으로 학습하거나 활용하지 않았음에도 불구하고, 대부분의 참여자가 여섯 개 범주를 일정 수준 재구성하였다는 점이다. 이는 참여자들이 동적 인터랙션을 무작위적으로 이해하는 것이 아니라, 실무 경험을 통해 형성된 암묵적 분류 체계를 가지고 있을 가능성을 시사한다.

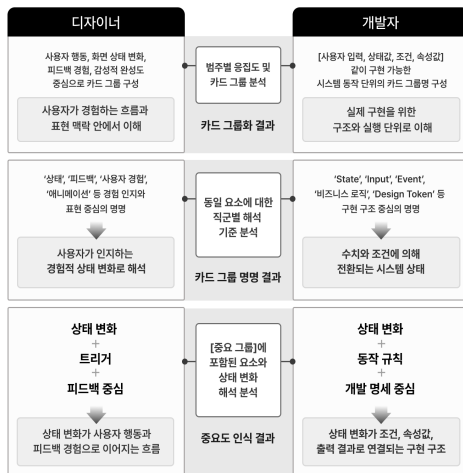
첫째, 카드 그룹화 결과, 두 직군 모두 의미적으로 유사한 항목을 동일한 그룹으로 묶는 경향을 보였다. 그러나 이러한 항목들을 해석하고 구조화하는 방식에서는 차이가 관찰되었다. 디자이너 참여자들은 혼합된 요소들을 사용자 경험의 흐름 속에서 움직임과 피드백을 조정하는 요소로 이해한 반면, 개발자 참여자들은 조건, 상태, 속성값, 출력 결과가 연결되는 구현 구조의 일부로 해석하였다. 즉 동일한 항목이 함께 분류되더라도, 이를 조직하고 의미화하는 방식은 직군에 따라 다르게 나타나는 것으로 확인되었다.

둘째, 카드 그룹 명명 결과에서는 직군별 분류 기준의 차이가 확인되었다. 디자이너 참여자들은 사용자 조작 행위, 화면 상태 변화, 피드백 경험, 브랜드 및 사용성 요소를 중심으로 동적 인터랙션을 조직하였으며, 이를 사용자 경험의 흐름 속에서 의미화하는 경향을 보였다. 반면 개발자 참여자들은 동일한 요소를 사용자 입력, 상태값, 조건, 속성값, 출력 결과 등 구현 가능한 시스템 동작 단위로 재구성하였다. 이는 본 연구의 제한된 표본 내에서 디자이너가 동적 인터랙션을 경험 중심의 개념 체계로 이해하는 반면, 개발자는 구현 중심의 개념 체계로 이해하고 있음을 시사한다.

셋째, 중요도 인식 비교 분석에서는 두 직군 모두 '상태와 모드'를 핵심적인 동적 인터랙션 요소로 인식하는 경향을 보였다. 그러나 해당 요소를 중요하게 여기는 관점에는 차이가 나타났다. 디자이너 참여자들은 상태를 사용자가 화면에서 인지하는 상황과 변화 과정으로 이해하였으며, 이를 트리거, 피드백 요소와 함께 경험의 흐름 속에서 해석하였다. 반면 개발자 참여자들은 상태를 컴포넌트가 관리해야 하는 상태값이자 상태 전이 구조로 이해하였으며, 상태 변화에 따른 동작 규칙과 구현 방식에 초점을 두고 해석하였다. 즉 동일한 상태 요소를 중요하게 인식하면서도 디자이너 참여자들은 경험 중심으로, 개발자 참여자들은 구현 중심으로 이해하는 경향을 보였다.

이러한 결과는 두 직군 간 차이가 특정 요소의 중요도 자체에 있기보다, 동일한 동적 인터랙션 요소를 어떠한 구조와 관계 속에서 이해하고 조직하는가에 있을 가능성을 보여준다. 디자이너와 개발자는 사용자 가치,

트리거, 상태와 모드와 같이 응집도가 높게 나타난 범주에서는 비교적 유사한 분류 구조를 형성하였으며, 이는 동적 인터랙션의 핵심 구성요소에 대해 직군 간 일정 수준의 공통된 인식 체계를 공유하고 있음을 시사한다. 이와 달리 동작 규칙, 피드백, 개발 명세와 같이 응집도가 상대적으로 낮은 범주에서는 카드의 배치와 범주화 방식에서 차이가 관찰되었다. 이러한 결과는 협업 과정에서 특정 구성요소 자체에 대한 이해 부족보다는, 해당 요소의 의미와 역할을 해석하고 다른 요소와 연결하는 방식의 차이로 인해 의사소통의 간극이 발생할 가능성을 보여준다. 특히 두 직군 모두 '상태와 모드'를 가장 중요한 요소로 인식하였다는 점은, 이 범주가 디자이너와 개발자 간 공통의 참조 기준이 될 수 있음을 시사한다. 이러한 결과는 [그림 7]과 같이 디자이너와 개발자가 동적 인터랙션의 주요 구성요소를 공통적으로 인식하면서도, 이를 연결하고 조직하는 방식에서는 차이를 보일 수 있음을 시사한다. 본 연구에서는 이러한 차이를 직군별 해석 구조의 차이로 설명하고자 한다.



[그림 7] 직군 간 인식 차이 종합

5. 논의 및 결론

본 연구는 동적 인터랙션의 전달 과정에서 나타나는 디자이너와 개발자 간 인식 차이를 카드소팅과 개념도 실험을 통해 비교·분석하였다. 연구 결과, 본 연구에 참여한 두 직군은 마이크로인터랙션의 구성 요소를 이해

하고 구조화하는 방식에서 차이가 관찰되었다.

디자이너 참여자들은 동적 인터랙션을 사용자 행동, 동작 규칙, 피드백 경험, 감성적 완성도와 연결된 경험 흐름으로 이해하는 경향을 보인 반면, 개발자 참여자들은 이를 상태값, 조건, 속성값, 구현 명세와 연결된 시스템 구조로 해석하는 경향을 보였다. 이러한 차이는 단순한 표현 방식이나 전달 오류의 문제가 아니라, 두 직군이 동일한 동적 인터랙션 요소를 이해하고 예측하는 멘탈모델 기반의 인식 차이에서 비롯될 수 있음을 보여준다. 이는 마이크로인터랙션 협업에서 동일한 용어와 요소를 공유하더라도, 참여자 내면의 해석 구조가 다를 수 있음을 시사한다. 따라서 동적 인터랙션의 설계와 구현 과정에서는 디자이너의 경험적 의도와 개발자의 구현 구조를 연결할 수 있는 공통 해석 틀을 마련하는 것이 중요하다.

본 연구의 주요 기여는 다음과 같다.

첫째, 동적 인터랙션 연구를 기능적 구조나 사용성 차원의 논의에서 확장하여, 디자이너와 개발자가 동적 인터랙션 요소를 어떠한 인식 구조로 이해하는지 멘탈모델 관점에서 탐색하였다.

둘째, 카드소팅과 개념도 실험을 통해 디자이너와 개발자의 분류 기준, 중요 요소 인식, 요소 간 관계 해석의 차이를 탐색적으로 규명하였다.

셋째, 이러한 인식 차이를 바탕으로 향후 AI Agent가 디자이너의 경험 의도를 개발자가 이해할 수 있는 형태로 변환하고 지원할 수 있는 해석적 협업 구조의 필요성과 가능성을 제시하였다.

본 연구의 한계와 향후 연구 방향은 다음과 같다.

첫째, 연구 참여자가 12명으로 제한되어 있어, 연구 결과를 모든 디자인·개발 협업 상황에 일반화하기에는 한계가 있다. 둘째, 참여자의 경력 수준, 조직 환경, 개인적 성향에 따라 동적 인터랙션을 이해하는 방식이 달라질 수 있으나, 본 연구에서는 이러한 개인 및 조직적 특성을 충분히 세분화하여 분석하지 못하였다. 셋째, 본 연구는 동적 인터랙션 요소를 중심으로 진행되었기 때문에, AI 서비스 기반 인터랙션이나 대화형 인터페이스 등 다른 유형의 인터랙션에서도 동일한 결과가 나타나는지는 추가 검토가 필요하다. 향후 연구에서는 더 다양한 직군, 경력 수준, 조직 환경을 포함하여 동적 인터랙션에 대한 인식 차이를 확장적으로 검토할 필요가 있다. 또한 본 연구에서 도출한 디자이너와 개발자의 멘탈모델 기반 인식 차이를 바탕으로, 경험 의도와 개발 명세를 연결하는 AI 기반 협업 지원 시스템의 구체적인 구조와 기능을 설계하고 검증할 필요가 있다.

참고문헌

1. Bétrancourt, M., & Boucheix, J.-M., Animation and motion in UX design. In Human-Computer Interaction, CRC Press, 2024.
2. Hassenzahl, M., Hedonic, emotional, and experiential perspectives on product quality. In Encyclopedia of human computer interaction, IGI Global Scientific Publishing, 2006.
3. Norman, D. A., The Design of Everyday Things (Revised and Expanded Edition), Basic Books, 2013.
4. Saffer, D., Microinteractions: Designing with details, O'Reilly Media, 2013.
5. Sharp, H., Jenny, P., & Rogers, Y., Interaction Design: Beyond Human-Computer Interaction, Aufl., Chichester, 2007
6. 김건동, "UI 컴포넌트 유형별 마이크로인터랙션의 트리거에 따른 피드백의 표현 분석", 디지털콘텐츠학회논문지, Vol. 25, No. 10, 2024.
7. 이동일, 이혜준, "소비자 집단 인터뷰에서 의미 네트워크 응집 구조의 이해", 소비자학연구, Vol. 23, No. 2, 2012.
8. Aagesen, P. T., & Heyer, C., & Personality of interaction: Expressing brand personalities through interaction aesthetics, In Proceedings of the 2016 CHI conference on human factors in computing systems, 2016.
9. Almaral Martinez, M. E., Berná-Martínez, J. V., Maciá Pérez, F., Lorenzo Fonseca, I., & Pascual Cáceres, C. P., "Understanding the importance of animation in human-computer interfaces and its relation to user experience", Proceedings of the 17th International Conference on Interfaces and Human Computer Interaction, ACM, 2023.
10. Arellano-Uson, J., Magaña, E., Morató, D., & Izal, M., "Survey on quality of experience evaluation for cloud-based interactive applications", Applied Sciences, 2024.
11. Betz, S., & Hall, R., "Self-archiving with ease in an institutional repository: Microinteractions and the user experience", Information Technology and Libraries, Vol. 34, No. 3, 2015.
12. Chen, Y., Shi, C., & Chen, L., "SpecifyUI: Supporting iterative UI design intent expression through structured specifications and generative AI", arXiv, 2025
13. Escamilla, A., Melenchón, J., Monzo, C., Morán, J. A., & Carrascal, J. P., "Exploring a user-centered approach for movement-based features in interaction design", International Journal of Human-Computer Studies, 2026.
14. Feng, K. J. K., Li, T. W., & Zhang, A. X., "Understanding collaborative practices and tools of professional UX practitioners in software organizations", Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023.
15. Forlizzi, J., & Battarbee, K., "Understanding experience in interactive systems", Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '04), ACM, 2004.
16. Goree, S., Doosti, B., Crandall, D., & Su, N. M., "Investigating the homogenization of web design: A mixed-methods approach", Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Article 196, 2021.
17. Kolte, A., & Rao, D., "Exploring Microinteractions in Human-Computer Interaction: Design Principles, Types, and User Experience", International Conference on Human-Centric Smart Computing, Springer Nature Singapore, 2024.
18. Lamine, Y., & Cheng, J., "Understanding

- and supporting the design systems practice", arXiv, 2022.
19. Leiva, G., Maudet, N., Mackay, W., & Beaudouin-Lafon, M., "Enact: Reducing designer-developer breakdowns when prototyping custom interactions", ACM Transactions on Computer-Human Interaction, 2019.
 20. Lenz, E., Diefenbach, S., & Hassenzahl, M., "Aesthetics of interaction: a literature synthesis", Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, ACM, 2014.
 21. Makkan, N., Brosens, J., & Kruger, R., "Designing for positive emotional responses in users of interactive digital technologies: A systematic literature review", HCI International 2020 - Late Breaking Papers: User Experience Design and Case Studies, Springer, 2020.
 22. Maudet, N., Leiva, G., Beaudouin-Lafon, M., & Mackay, W. E., "Design breakdowns: Designer-developer gaps in representing and interpreting interactive systems", Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17), ACM, 2017.
 23. Merz, B., Tuch, A. N., & Opwis, K., "Perceived user experience of animated transitions in mobile user interfaces", Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 2016.
 24. Rugg, G., & McGeorge, P., "The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts", Expert Systems, 2002.
 25. Schrepp, M., Rosenzweig, E., Soares, M. M., & Marcus, A., "A review of design, user experience, and usability in HCI, through the lens of the DUXU conference", International Journal of Human-Computer Interaction, 2025.
 26. Sousa, V., Syriani, E., & Fall, K., "Operationalizing the integration of user interaction specifications in the synthesis of modeling editors", Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering, 2019.
 27. Tchivi, E., Sharma, B., & Paea, S., "A systematic review of the comparison of different types of card sorting", IEEE Access, 2025.
 28. Van Der Veer, G. C., Lenting, B. F., & Bergevoet, B. A., "GTA: Groupware task analysis-Modeling complexity", Acta psychologica, 1996.
 29. Xu, W., "Maximizing the value of enterprise human-computer interaction standards: Strategies and applications", arXiv, 2021.
 30. Zhang, S., Zhang, T., Cheng, J., & Zhou, S., "Who is to blame: A comprehensive review of challenges and opportunities in designer-developer collaboration", Proceedings of the ACM on Human-Computer Interaction, 9(2), 2025.
 31. developer.apple.com
 32. m3.material.io
 33. www.hiixlab.works