

# 협업 환경에서의 일관성 확보를 위한 GPT 기반 코딩 프로세스

정한민<sup>1,2</sup>, 박정훈<sup>1</sup>, 유수현<sup>1</sup>  
<sup>1</sup>한국과학기술정보연구원 융합서비스센터  
<sup>2</sup>과학기술연합대학원대학교 응용AI전공  
 {jhm, jhpark, yoos}@somewhere.ac.kr

## GPT-based Coding Process for Consistency in a Collaborative Environment

Hanmin Jung<sup>1,2</sup>, Jung Hoon Park<sup>1</sup>, Suhyeon Yoo<sup>1</sup>  
<sup>1</sup>Convergence Service Center, Korea Institute of Science and Technology Information  
<sup>2</sup>Applied AI Major, University of Science and Technology

### 요 약

본 연구는 프로그래밍 협업 환경에서 생성형 AI인 ChatGPT-4를 활용한 코딩 프로세스를 제안한다. 일관성 있는 결과를 얻기 위해 프롬프트 생성, GPT 실행, 의사코드 변환, 코드 비교, 동일 코드 생성 여부 판단, 테스트 실행, 동일 결과 생성 여부 판단, 코드 검사 및 수정의 8단계를 거친다. 팀 프로젝트와 페어 프로그래밍 등의 다양한 협업 환경에 적용 가능한 이 프로세스를 통해 생성형 AI를 효과적으로 활용할 수 있음을 보여주었다는 점에서 그 의미가 있다. 본 연구는 생성형 AI를 활용한 협업 환경에서의 코딩이 본격적으로 이루어질 것으로 예상되는 이 시점에서, 인간-AI 협업 환경에서의 코딩 효율성 및 일관성을 높일 수 있을 것으로 기대한다. 이러한 연구는 인간과 AI가 함께 작업하는 미래를 위한 기초를 마련하는 데 중요한 역할을 할 것이다.

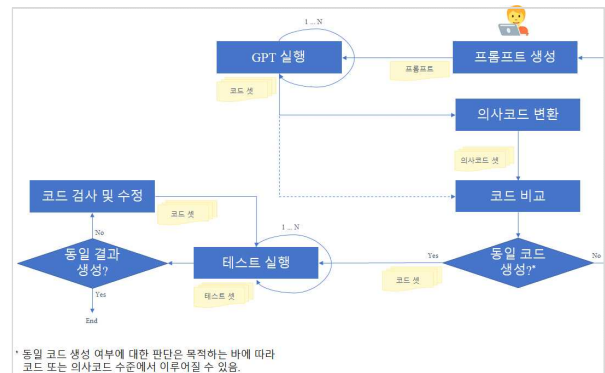
### 1. 서론

현재 시대는 프로그래밍 규모가 혼자 전담하기 어려울 정도로 커져서, 프로그래밍을 위한 협업 환경에 대한 고민이 커지고 있다. 이러한 협업 환경은 대규모 팀 프로젝트에서 동일한 코드를 공유하며 업데이트하는 방식과 페어 프로그래밍(Pair Programming)과 같이 두 프로그래머가 내비게이터(Navigator)와 드라이버(Driver) 역할을 수행하는 방식으로 나눌 수 있다. 전자의 경우 협업 참여자들이 동일한 코드를 공유하며 버전 관리를 해야 하고, 후자의 경우 둘 간의 대화가 명확해야 한다. 특히, 페어 프로그래밍 관련하여 협업 환경에서 원활한 코딩을 위한 협업 도구를 이용한 교육을 제안하는 연구 [1]와 페어 프로그래밍 환경에서의 협업 효과를 분석하는 연구 [2] 등이 주요 연구 방향이다. 실제 필드에서는 깃(Git) 등 도구를 이용하여 개발자들이 차례로 수정한 다음 최종 산출물을 하나의 코드 베이스로 합치거나 라이브 협업이라 하여 실시간에 다수가 동일 코드를 공유하고 편집하면서 개발 시간을 단축하는 방식으로 협업을 진행하는 경우가 많다.

그러나 최근에는 ChatGPT와 같은 생성형 AI의

코딩 능력이 발달함에 따라 인간-인간 협업이 아닌 인간-AI 간 협업에 관한 연구가 점점 중요해지고 있다. 본 연구는 대규모 팀 프로젝트나 페어 프로그래밍 환경에서 인간 일부가 AI로 대체되는 가까운 미래에 어떤 이슈가 있을지, 그리고 AI를 활용한 코딩을 어떻게 수행할 수 있는지를 실제 예시를 통해 보여주고자 한다. 특히, 코딩 결과의 일관성 확보를 위한 프로세스를 제안하며, 프롬프트 설계부터 코드 생성까지의 과정을 안내하는 데 초점을 맞춘다.

### 2. 협업 환경에서의 GPT 기반 코딩 프로세스



(그림 1) GPT 기반 코딩 일관성 확보 프로세스

생성형 AI는 동일 프롬프트에 대해 이질적인 코딩 결과를 만들어 내는 특징이 있어서, 동일한 코드를 공유하고 갱신해야 하는 협업 환경에 바로 적용하기에 어려움이 있다. 이에 본 연구는 팀 프로젝트, 페어 프로그래밍 등 협업 환경에서의 일관성을 확보하기 위한 GPT 기반 코딩 프로세스를 다음과 같이 정의하고, 각 단계에 대한 설명을 제공하고자 한다 ((그림 1) 참조).

1. 프롬프트 생성: 최초 프롬프트는 달성하고자 하는 목적, 예외 상황, 조건을 포함하여 작성되어야 한다[3]. 다음은 본 연구에서 사용한 실제 프롬프트 예시이다[4].

최초 프롬프트: {Write a program in C code that calculates the three angles within a triangle given the lengths of the three sides of the triangle. Use of only main function. Apply the triangular inequality principle, but also check that the length of each side is positive.}<표 1> 참조)

2. GPT 실행: 주어진 프롬프트를 입력으로 GPT (본 연구에서는 GPT-4 이용)를 주어진 횟수 N번 실행한다<sup>1)</sup>. N의 크기는 얼마나 일관성의 중요도에 따라 결정되며, 본 연구에서는 충분한 비교를 위해 N을 10으로 설정한다. 이는 동일한 프롬프트로 생성된 다수의 코드를 비교하기 위함이며, 코드 간 이질성을 찾는 데 목적이 있다(<표 1> 참조).

3. 의사코드(Pseudo Code) 변환: 생성된 다수의 코드를 그 자체로 비교하는 경우 로직의 일관성을 판단하기 어려우므로, 의사코드로 변환하여 추상화된 상태에서 비교할 필요가 있다(<표 2> 참조).

4. 코드 비교: 코드 비교는 생성 코드 수준에서의 비교(예. 변수명 등)와 로직 수준에서의 비교(의사코드 간 비교)로 나누어 볼 수 있으며, 목적에 따라 프로젝트를 수행하는 팀에서 결정하면 된다.

5. 동일 코드 여부 판단: 코드 간 비교를 통해 모든 경우(N개)에 대해 동일 코드가 생성되었다고 판단되면 테스트 실행으로 넘어가며, 그렇지 않은 경우에는 프롬프트를 수정하여 1번 단계부터 다시 반복한다. 다음은 본 연구에서 사용한 최종 프롬프트 예시이다.

최종 프롬프트: 최초 프롬프트 + {separately as

exceptional condition.}<sup>2)</sup> {Name the variables for the three angles as *\_a*, *\_b*, and *\_c*, and the variables for the three sides as *a*, *b*, and *c*.}<sup>3)</sup> {Take the lengths of three sides from user at once.}<sup>4)</sup> {Calculate the three angles in degrees at once.}<sup>5)</sup> {Finally, print the three angles one by one.}<sup>6)</sup>

6. 테스트 실행: 프롬프트 내 예외 상황, 조건을 모두 테스트할 수 있는 테스트 셋을 준비하여 실행한다.

7. 동일 결과 생성 여부 판단: N개의 코드에 대해 동일한 테스트 셋으로 테스트한 결과가 모두 동일함을 판단하며, 동일한 경우에는 종료 단계로 넘어가 해당 코드를 버전 관리 대상으로 하고, 그렇지 않은 경우에는 코드를 검사해야 한다.

8. 코드 검사 및 수정: 코드 내 로직과 문장 (Statement)들을 검사하여 프롬프트에서 요구한 내용을 정확히 구현하였는지를 검사하고 필요한 경우, 수정해야 한다.

### 3. 결론

본 연구<sup>7)</sup>는 팀 프로젝트와 페어 프로그래밍 등 다양한 협업 환경에서 ChatGPT와 같은 생성형 AI의 동적 변환 능력을 프롬프트를 이용하여 어떻게 통제하면서 코딩 협업을 수행할 수 있을 것인지에 대한 프로세스를 정립하였다. 향후 본격적으로 생성형 AI를 활용한 협업 환경에서의 코딩이 이루어질 것으로, 프롬프트 공학(Prompt Engineering)을 포함한 다양한 세부 영역에서의 후속 연구를 기대된다.

### 참고문헌

- [1] Y. Kim and S. Chun, "SW Education Program using Pair Programming Collaboration Tools," *Journal of the Korean Association of Information Education* 23(4), 2019.
- [2] J. Hannay, T. Dybå, E. Arisholm, and D.

<sup>2)</sup> 구문을 분리하여 예외 상황을 처리

<sup>3)</sup> 변수명을 통제

<sup>4)</sup> 입력값들을 받는 방식을 제약

<sup>5)</sup> 구해야 할 해답들을 계산하는 방식을 제약

<sup>6)</sup> 해답들을 출력하는 방식을 제약

<sup>7)</sup> 본 연구는 KISTI 주요사업인 K-23-L01-C04-S01 지능형 과학 기술지식인프라 통합서비스 플랫폼 구축 사업의 하나로 실시됨.

<sup>1)</sup> 프롬프트로부터 자동 생성된 GPT-4 코드 예시는 <표 1> 참조

Sjøberg, “The effectiveness of pair programming: A meta-analysis,” Information and Software Technology 51(7), 2009.

[3] H. Jung and J. Park, “Design and Issues of Writing Literatures Using ChatGPT,” Journal of Knowledge Information Technology and Systems 18(1), 2023.

[4] H. Jung, J. Park, and S. Yoo, “Programming with ChatGPT in terms of Completeness and Soundness,” Proceedings of Korean Society for Internet Information Spring Conference, 2023.

<표 1> 최초 프롬프트로부터 자동 생성된 GPT-4 코드 예시

```
#include <stdio.h>
#include <math.h>

int main() {
    float a, b, c;
    float A, B, C;
    printf("Enter the lengths of the three sides of the triangle: ");
    scanf("%f %f %f", &a, &b, &c);

    // Check that the lengths of the sides are positive
    if (a <= 0 || b <= 0 || c <= 0) {
        printf("Error: sides must have positive length.\n");
        return 1;
    }
}
```

```
// Check the triangular inequality
if (a + b <= c || a + c <= b || b + c <= a) {
    printf("Error: sides violate the triangular inequality.\n");
    return 1;
}

// Calculate the angles using the law of cosines
A = acos((b*b + c*c - a*a) / (2*b*c));
B = acos((c*c + a*a - b*b) / (2*c*a));
C = acos((a*a + b*b - c*c) / (2*a*b));

// Convert from radians to degrees
A = A * 180.0 / M_PI;
B = B * 180.0 / M_PI;
C = C * 180.0 / M_PI;

printf("The three angles of the triangle are: %f, %f, %f\n", A, B, C);

return 0;
}
```

<표 2> 프롬프트, 대응 의사코드, 변이 의사코드 예시

Prompt	Corresponding Pseudo Code	Variations
Write a program in C code	Include external libraries (stdio.h, math.h)	
Use of only main function	main() {	
	Define variables (a, b, c, _a, _b, _c)	1. a, b, c, angleA, angleB, angleC 2. sideA, sideB, sideC, angleA, angleB, angleC 3. side_a, side_b, side_c, angle_A, angle_B, angle_C
given the lengths of the three sides of the triangle	Take the lengths of three sides from user one by one	1. at once
Apply the triangular inequality principle, but also check that the length of each side is positive	Check if the lengths of the three sides are not positive Check if the triangular inequality principle is not satisfied	1. Check if the lengths of the three sides are positive and the triangular inequality principle is satisfied 2. Check if the lengths of the three sides are positive and the triangular inequality principle is satisfied (nested if)
calculates the three angles within a triangle	Calculate the three angles in radians Convert the three angles from radians to degrees	1. Calculate the three angles in degrees
	Print the three angles at once	1. one by one
	}	