

## 기호학적 관점에서 프로그래밍 사고의 이해

김동만<sup>o</sup>, 이태욱<sup>\*</sup>

<sup>o</sup>한국고원대학교 컴퓨터교육과,

<sup>\*</sup>한국고원대학교 컴퓨터교육과

e-mail: emotionman@indischool.com<sup>o</sup>, twlee@knue.ac.kr<sup>\*</sup>

## Understanding of programming thinking from Semiotics Perspective

Dong Man Kim<sup>o</sup>, Tae Wuk Lee<sup>\*</sup>

<sup>o</sup>Dept. of Computer Education, Korea National University of Education,

<sup>\*</sup>Dept. of Computer Education, Korea National University of Education

### ● 요약 ●

이 연구의 목적은 기호학적 관점에서 프로그래밍에서 발생하는 학습자의 사고 과정을 이해하기 위함이다. 그래서 프로그래밍의 표상과정을 이해하기 위한 기호작용 모형을 제안하였다. 이 연구의 결론은, 프로그래밍 교육에서 구성주의(constructivism) 학습 이론을 적용하기 위해서는 개인의 해석체와 프로그래밍 요소에서 인터텍스트(intertext) 속성을 파악하는 것이 선결과제인 것과 프로그래밍 맥락인 콘텍스트(context)의 중요함을 확인하였다. 후속 연구로 인지언어학적 방법으로 학습자가 프로그래밍에서 표상한 해석체(interpretant)와 콘텍스트(context), 인터텍스트(intertext) 등의 상호작용을 구체적으로 알아보는 연구를 진행하고자 한다.

**키워드:** Programming Thinking, Semiotics, Symbolic Model at Programming, Constructivism

### I. Introduction

학습자 중심의 효과적인 프로그래밍 교육을 진행하기 위해서는 학습자가 어떤 사고과정을 통해 프로그래밍을 하는지 이해할 필요가 있다. 학습자는 프로그래밍에서 필요한 정보를 머릿속에서 적절히 조합하고 활용하여 결국 프로그램으로 완성한다. 이 과정은 고등사고력을 필요로 하는 복잡한 인지 과정이다. 그래서 학습자가 프로그래밍에 필요한 다양한 정보로 질서를 조직하여 코드로 구체화시키는 복잡한 인지 과정을 이해하기 위해, 기호학적 관점에서 접근해볼 수 있다. 왜냐하면, 학습자가 프로그래밍 할 내용이나 설계를 읽고 필요한 정보를 머릿속에 떠올리는 것은 기호 해석의 결과이기 때문이다. 이 연구는 기호학적 관점에서 프로그래밍에서 발생하는 인지 과정을 이해하기 위함을 목적으로 기호이론을 바탕으로 학습자가 프로그래밍 과정에서 머릿속에서 발생하는 일을 구체적으로 설명하고자 한다.

참조한다. Peirce는 기호의 작용을 인지적 표상의 관점에서 접근하여, 기호(sign)를 사람이 대상(object)을 해석할 수 있게 하는 매개로 보았다 [1]. 그는 사람이 기호를 보고 대상 자체를 떠올릴 수 없기 때문에 기호가 지시하고 있을 것이라고 예상되는 해석체(interpretant)를 떠올리는데, 이런 작용을 표상(representamen)이라 하였다. Witte(1992)는 피스의 이론을 바탕으로 쓰기에 기호작용 모형을 제시하면서, 필자의 텍스트 구성이 텍스트(text), 콘텍스트(context), 인터텍스트(intertext) 등의 상호작용으로 이루어진다고 하였다[1]. 그리고 그는 필자가 텍스트로 드러내려는 것은 콘텍스트이며, 이것은 필자가 텍스트로 표현하려는 내용과 관련된 생각으로 필자가 텍스트를 구성하기 위하여 머릿속에 떠올리고 고려해야 할 여러 요인이라고 하였다[2]. 그래서 콘텍스트에 의하여 텍스트가 구성되고, 텍스트의 내용과 표현을 결정한다. 필자는 콘텍스트를 고려하여 텍스트를 구체화하는 과정에서 인터텍스트를 활용하게 된다. 인터텍스트는 두 텍스트 이상의 텍스트들이 특정 내용을 공유하는 것을 가리킨다[2]. 즉 필자가 텍스트를 생성하는 과정은 콘텍스트를 구체화하기 위해 관련 정보들인 인터텍스트를 결합하는 사고 과정이다.

### II. Symbolic Model

프로그래밍의 사고의 결과물은 프로그램이 작성되는 쓰기 과정이다. 그래서 이 논문에서는 학습자의 프로그램 사고 과정의 해석을 위해 Peirce의 기호 이론을 바탕으로 Witte의 쓰기에서 기호 작용 모형을

### III. Understanding Programming Thinking and Symbolic Model at Programming

Witte(1992)는 우리가 장보기 메모를 작성할 때, 어떤 요리를 할 것이고, 누가 먹는지, 방문할 마트와 마트에 있는 물건이 무엇인지까지도 고려하는 것을 콘텍스트라고 했다[2]. 그래서 프로그래밍에서 콘텍스트는 프로그래밍할 내용을 구성해야 하는 상황과 조건, 즉 컴퓨터와 사용자를 포함한 디지털 컴퓨팅 환경을 고려하는 것이다. 학습자가 프로그램을 구성하기 위해서는 해석체와 내용 공유가 반드시 있어야 한다. 위트 설명에 따르면 주부의 장보기 목록 메모도 마트에 있는 상품명을 써야 하기 때문에 인터텍스트를 이룬다[2]. 프로그래밍 활동에서는 프로그래밍 구성요소가 인터텍스트를 이룬다.

학습자는 프로그램의 작동 과정을 대상(object)으로 두고 인터텍스트를 매개로 코드(code), 즉 텍스트를 구성하는 활동이다. 그래서 프로그램은 학습자가 인터텍스트들에 질서를 부여하여 구성된 텍스트의 집합이다. 학습자는 프로그램의 작동 과정인 대상을 보고 정보를 표상해야 한다. 학습자가 읽고 표상한 내용이 해석체이다. 학습자가 해석체를 구성하는 활동은 텍스트의 기호를 단서로 프로그램 작동 과정을 표상하는 콘텍스트를 구체화할 코드를 표상하는 사고 작용이다. 학습자는 대상을 보면서 질서를 부여해 놓은 많은 해석체를 만났고 동시에 프로그래밍에 필요한 인터텍스트를, 표상된 해석체 밖에서 끌고 온다. 학습자의 머릿속에는 프로그래밍 할 내용의 해석체와 프로그래밍의 인터텍스트가 콘텍스트와 관련되면서 떠오르게 된다. 결국 학습자는 해석체들과 인터텍스트들을 머릿속에 표상된 정보가 많아지면서 복잡함(complex)을 경험한다.

한편, 프로그래밍을 할 때 학습자가 프로그램으로 드러내려는 것은 대상인 콘텍스트는 학습자가 프로그램을 구성하기 위하여 머릿속에 떠올리고 고려할 여러 맥락이다. 그래서 콘텍스트는 컴퓨터 프로그램의 내용과 표현을 결정한다. 학습자는 콘텍스트를 고려하여 프로그램을 구체화하는 과정에서 인터텍스트를 활용하게 된다. 학습자가 프로그램 내용 구성의 인터텍스트를 프로그래밍 구성 요소로 구현하는 것은 컴퓨팅 환경을 이용하기에 당연히 필요한 인지 과정이다. 그래서 컴퓨터 프로그램은 콘텍스트를 구현하기 위해 인터텍스트를 활용한 코드로 표현된다. 그리고 학습자가 갖고 있는 인터텍스트는 대상을 떠올리게 하고, 다시 이와 관련된 콘텍스트를 생성하고, 이 콘텍스트에서 인터텍스트가 다시 생겨난다. 이처럼 학습자가 프로그램을 생성하는 사고 과정은 콘텍스트를 구체화하기 위해 관련 정보들인 인터텍스트를 결합하는 인지 과정이다. 이런 지속적 과정을 거치고 나서 콘텍스트를 구현할 수 있는 충분한 인터텍스트를 결합하면 이에 필요한 프로그램 코드를 더 연결하기 위해 콘텍스트를 확인하고 다시 인터텍스트를 결합하는 과정을 반복한다. 학습자는 여러 인지 과정으로 갖게 된 해석체들과 인터텍스트들이 복잡하게 뒤엉킨 사고 공간에서 콘텍스트를 고려하여 질서를 부여함으로써 프로그램 코드로 구성되는 기호작용을 경험한다. 이와 같이 학습자가 프로그래밍에서 겪는 기호작용을 모형으로 표현하면, Fig.1.과 같이 나타낼 수 있다.

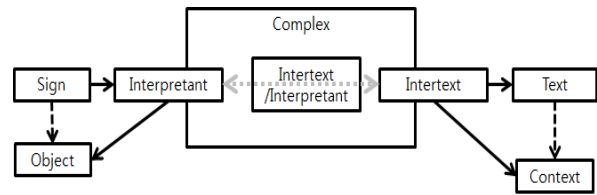


Fig. 1. Symbolic Model at Programming

프로그래밍에 참여한 학습자가 경험하는 인지 과정을 기호작용 모형을 통해 설명하자면, 학습자는 프로그램 작동 과정인 대상(object)를 고려하면서 프로그램 설계나 의사코드(sign)를 해석(interpretant)하여 프로그램을 구성할 표상이 머릿속에서 인터텍스트(intertext)를 만나야 프로그래밍 코드, 즉 텍스트(text)를 생성한다. 이어서 학습자는 머릿속에 해석체와 프로그램에 필요한 프로그래밍 요소인 인터텍스트가 복잡하게 존재하다가 컴퓨팅 맥락인 콘텍스트(context)를 고려하면서 선택된 인터텍스트가 확정되면, 이를 나타낼 기호인 텍스트로 표현한다.

### IV. Conclusions

이 논문은 기호학적 관점에서 프로그래밍 사고에 대해 알아보고 프로그래밍의 인지 과정을 이해하기 위한 모형을 제안하였다. 이 연구로 도출된 결론은, 프로그래밍 교육에서 구성주의 학습 이론을 적용하기 위해서는 개인의 해석체와 프로그래밍 요소에서 인터텍스트적 속성을 파악하는 것이 선결과제임을 확인하였다. 또한 프로그래밍 맥락인 콘텍스트의 중요함을 확인하였다. 후속 연구로 학습자가 프로그래밍에서 표상한 해석체와 콘텍스트, 상호텍스트 등이 어떻게 유기적인 상호작용하는지 인지언어학적 관점에서 구체적으로 알아보는 연구를 진행하고자 한다.

### REFERENCES

[1] Witte. S. P., "Context, Text, Intertext: Toward a Constructivist Semiotic of Writing," Written communication, Vol. 9, No. 2, 237-308, Apr. 1992.  
 [2] Kim, D. N., "Intertextuality and text understanding education," seoul : bagijeong, Sep. 2014.