

# TMO 기반의 실시간 소프트웨어 개발을 위한 지원 도구 설계

오철<sup>0\*</sup> 신원\* 김태완\*\* 장천현\*

\*건국대학교 컴퓨터공학과

\*\*명지대학교 전기공학과

ironism@Konkuk.ac.kr, wonjjang@Konkuk.ac.kr, twkim@mju.ac.kr, chchang@Konkuk.ac.kr

## Design of Supporting Tool for TMO-based Real-time Software Development

Cheol Oh<sup>0\*</sup> Won Shin\* Taewan Kim\*\* Chunhyon Chang\*

\*Dept. of Computer Science and Engineering, Konkuk University

\*\*Dept. of Electrical Engineering, Myongji University

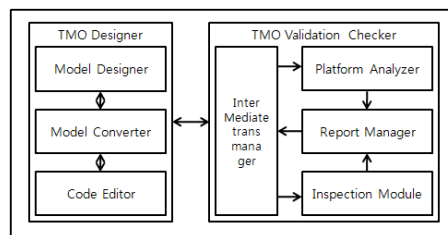
### 1. 서론

실시간 소프트웨어는 비행기 제어, 원자력 발전소 제어 등 고신뢰성 분야에 사용된다. 이와 같은 실시간 소프트웨어는 연산 결과의 정확성과 결과 처리의 시간적인 정확성을 지켜주지 못하면 막대한 경제적 피해를 주거나 인명사고로 이어 지기도 하여 높은 신뢰성을 요구하게 된다. 그러므로 정해진 시간 내에 작업을 완료하는 시간의 정확성에 대한 연구는 현재까지 중요한 이슈이다. 그 결과 중 하나로 Time-triggered Message-triggered Object (TMO) 모델이 제시 되었다[1]. TMO 모델은 실시간 개념에 따른 시간 처리의 다양한 기능을 지원하고 적시성을 보장하여 실시간 소프트웨어 개발에 많은 도움이 되었다. 게다가 TMO 모델 기반 실시간 소프트웨어의 개발을 돕고자 GUI를 지원하는 Visual Studio for TMO (ViSTMO)가 등장하여 소프트웨어 개발을 수월하게 하였다[2]. 하지만 ViSTMO는 다양한 플랫폼에 대하여 적용하기 위한 플랫폼 독립성에 관한 문제점과 다양한 언어와 많은 코드 량으로 인한 코드 검증에 어려움을 주는 안전성 문제가 있다.

이에 본 논문에서는 TMO 모델 기반 실시간 소프트웨어의 개발에 있어 다양한 플랫폼이 적용 가능한지에 대해 분석하여 주고 개발 단계에서부터 오류를 검증하고 개발 작업 중 이를 반영해 소프트웨어 개발에 도움을 줄 수 있는 TMO Studio를 제안한다. TMO Studio는 모델링 작업을 담당하는 개발도구와 오류 검증도구를 유기적으로 동작하여 개발의 편의를 돕고 나아가 기간단축과 비용절감의 이점을 줄 수 있다. 또한, 플랫폼의 범용성을 향상시켜 다양한 개발환경에서 사용할 수 있도록 도와준다.

### 2. 본론

TMO Studio는 TMO 모델 기반의 실시간 소프트웨어 개발 과정에서 편리하고 안전성 있는 개발을 할 수 있도록 도와준다. 이는 개발 단계부터 오류 정보를 반영할 수 있는 오류 검증 방법과 다양한 플랫폼을 대상으로 개발 할 수 있게 도와주는 플랫폼 분석 방법을 개발 작업 도구에 반영하는 범용성과 안전성을 보장할 수 있는 개발 도구이며 TMO Studio는 역할에 따라 개발 작업을 할 수 있는 TMO Designer와 여기에 도움을 주는 Validation Checker로 나뉜다.



< TMO Studio 전체 구조 >

TMO Designer는 개발 작업 도구로써 모델링을 하기 위한 개발 Model Designer와 모델링의 결과를 코드로 변환, 생성 시켜 주는 Model Converter, 그리고 코드 편집 작업을 위한 Code Editor로 구성되어있다. TMO Designer는 TMO 모델 기반 실시간 소프트웨어 개발에 시각적인 도움을 주어

개발을 수월하게 한다. 또 메타 기반의 개발로써 템플릿 형태로 미리 설계되어 있는 코드를 계속 사용할 수 있다. 그러므로 사용자는 효율적인 코딩을 할 수 있고 불필요한 코드 개발을 막을 수 있는 장점을 가진다.

Intermediate Language Trans Manager (ILTM)는 TMO Studio에서 생성된 소스 코드를 Platform Analyzer와 Inspection Module에서 각각 사용할 수 있는 Intermediate Language data (IL data)로 변형한다. 다시 말해서 메타 데이터로 이루어진 소스코드가 TMO Designer에서 ILTM으로 넘어오면 사용자가 작성한 소스 코드만 추출하고 AST 생성 모듈의 어휘 분석과 구문 분석을 통하여 Abstract Syntax Tree (AST)로 생성한다. 이러한 AST를 IL data 생성 모듈을 통하여 Platform Analyzer와 Inspection Module이 필요로 하는 형식으로 구성된 IL data로 생성한다. IL data는 각 도구에서 필요로 하는 정보를 효율적으로 처리하기 위해 데이터를 읽어 오는 규칙에 따라 한 줄 단위로 읽어오는 Single Block과 뭉쳐있는 일련의 코드를 읽어오는 Mass Block, 그리고 여러 곳에 나누어져 존재하는 코드를 읽어오는 Multi Block으로 나누어 생성된다.

플랫폼 영향도 분석을 위한 Platform Analyzer는 넘어온 IL data를 사용하여 다양한 플랫폼에 적용하기 위한 분석을 한다. 이 과정은 사용자가 만든 소스코드가 표준코드로 변환 가능한지 비교한 후, 가능하다고 판단된 소스코드를 가지고 다시 플랫폼 관련 정보를 담고 있는 저장소에 적용시켜 결과를 알아내는 것으로 IL data는 Platform Analyzer의 표준코드 검사 모듈과 이기종 플랫폼 변환 가능 확인 모듈을 지나면서 비교 및 분석 된다. 이를 통해 도출된 결과는 Report Manager를 경유하여 TMO Designer에 전달되고 개발 과정에 반영하여 플랫폼 범용성을 향상 시키고 다양한 개발환경에서도 사용할 수 있는 소프트웨어를 만들 수 있다.

Inspection module은 전달된 데이터를 대상으로 코딩 룰과 취약점 분석을 통해 프로그램의 논리적 오류를 줄여준다. Inspection module에 전달된 IL data는 오류패턴검색 과정을 통해 검색된 오류 패턴에 따라 코딩 룰 검색 모듈이나 취약점 분석 모듈, 또는 양쪽 모두로 전달된다. 전달된 데이터를 코딩 룰 검색 모듈에서 검색하여 코딩 룰에 반하거나 문제 될 수 있는 부분을 찾아내고 그 결과 값을 Report Manager에 보낸다. 또한 취약점 분석 모듈로 전달된 데이터를 취약점 정보를 통해 비교 및 분석 처리하고 그 결과 값이 Report Manager를 통하여 TMO Designer에게 전달되고, 안전성 있는 실시간 소프트웨어의 개발을 할 수 있도록 도와준다.

TMO Studio는 Designer 도구와 Validation checker와의 유기적인 관계를 형성하고 있어 개발 단계부터 안전성을 고려할 수 있는 장점을 가지고 있다. 그리고 개발된 소프트웨어의 범용성을 위해 플랫폼 적용 여부를 가능하고 나아가 다양한 플랫폼에 사용할 수 있게 도와준다. 또한 Designer에서 넘어오는 메타 데이터를 IL data로 변환하는 과정에서 각 모듈에 필요로 하는 IL data 형식이 다르기 때문에 사용할 분석모듈에 모두 적용될 수 있는 데이터 구조를 설계하여 분석 중에 효율적인 검색을 할 수 있도록 도와준다. 그리고 코딩 룰 정보와 취약점 정보를 담고 있는 저장소를 보완하고 확장하여 논리적 정확성을 높일 수 있다.

### 3. 결론

실시간 소프트웨어는 정해진 시간 안에 작업을 수행해야 하는 것이 가장 중요하다. 이를 위한 연구로써 TMO 모델이 제시되었고 TMO 모델 기반의 실시간 소프트웨어 개발을 돕기 위한 방법 중 하나로 GUI를 지원하는 ViSTMO가 등장하였지만 ViSTMO는 플랫폼 범용성에 관한 지원과 소프트웨어 안전성에 대한 지원이 미흡하였다.

본 논문에서는 TMO 모델 기반의 실시간 소프트웨어의 개발에 있어서 Platform Analyzer로 이기종 플랫폼 적용 여부를 분석하고 Inspection Module을 통하여 개발 단계에서부터 오류를 검증하여 개발에 도움을 주는 TMO Studio를 제안하였다. 이를 통해 플랫폼 범용성을 향상시키고 다양한 개발환경에서도 사용할 수 있는 소프트웨어를 만들 수 있다. 게다가 개발 단계부터 오류 검증을 반영한 소프트웨어를 개발하여 개발 기간을 단축하고 비용을 절감 할 수 있으며 무엇보다도 실시간 소프트웨어의 논리적 안전성을 향상 시킬 수 있다.

향후 연구 과제로는 본 논문에서 제안한 TMO Studio의 설계를 보완하고 구현할 예정이다.

### 참고문헌

- [1] K.H. Kim, "Object-Oriented Real-Time Distributed Programming and Support Middleware," 7th IEEE ICPADS, pp.10-20, 2000
- [2] K.H. Kim, "A GUI Approach to Programming of TMO Frames and Design of Real-Time Distributed Computing Software" IEEE CS Press, pp. 53-60, 2003