

모델 변환을 이용한 윈도우 모바일 어플리케이션 개발*

손현승* 김우열* 김재승** 김영철*[○]

홍익대학교 컴퓨터정보통신공학과* 트라이콘**

son@selab.hongik.ac.kr john@hongik.ac.kr ceo@tricon.co.kr bob@hongik.ac.kr

Development of Windows Mobile Application using Model Transformation Technique

Hyun-seung Son* Woo-Yeol Kim* Jea-seung Kim** R. Young-chul Kim*[○]

Dept. of CIC, Hongik University, Jochiwon, Korea* TRICON**

1. 서 론

소프트웨어 개발을 빠르게 하기 위해서는 기존에 만들었던 자원들을 최대한 활용 가능한 재사용 체계가 갖추어져 있어야 한다. 그러나 모바일 임베디드 소프트웨어는 시스템에 종속적인 특성을 지니고 소스 코드를 중심으로 개발이 진행되기 때문에 소프트웨어의 재사용이 어렵다. 더욱이 각 휴대폰 제조 및 공급 회사마다 다양한 플랫폼을 제공하기 때문에 이를 해결할 수 있는 방법이 필요하다.

MDD(Model Driven Development)는 플랫폼에 독립적인 메타모델을 설계한 후 필요한 기술 모델을 변경하여 그 모델을 통해 코드 생성을 자동화하는 메커니즘이다. 만약 응용 프로그램이 다른 구현 환경에서 필요하다면 그 환경에 대한 모델을 선택하고 다시 코드를 생성하면 된다. 이때 응용 프로그램 모델을 수정할 필요는 없다. 이와 같은 방법으로 모델의 재사용과 관련된 코드의 생산성을 높일 수 있다 [2-5]. 즉, 스마트폰 개발환경에 모델변환기법을 적용하면 하나의 모델을 이종의 모델로 자동생성이 가능하다. 본 논문에서는 이종의 스마트폰 개발을 위해서 윈도우 모바일 플랫폼 개발에 모델 변환 기법을 적용한다. 윈도우 모바일 플랫폼에 적용하기 위해서 어플리케이션의 구조를 분석하고 독립적인 특성과 종속적인 특성들로 분류 변환한다. 윈도우 모바일 플랫폼으로 모델 변환을 위한 규칙들을 제안한다. 모델 변환의 규칙은 모델 변환 언어인 ATL(ATLAS Transformation Language)으로 기술한다. 변환 과정에 사용 되는 타겟 독립 모델(TIM)은 플랫폼에 의존적이지 않은 모델을 의미하고 타겟 종속 모델(TSM)은 플랫폼에 의존적인 것을 말한다 [2].

2. 본론

모델 변환은 입력되는 소스 모델을 대상 모델로 변환하는 방법이다. 모델 변환의 종류에는 모델에서 모델, 모델에서 코드, 코드에서 모델 등이 있다. 또한 모델 변환에서 사용되는 모델은 UML 뿐만 아니라 컨트롤 플로우 다이어그램, 데이터 플로우 다이어그램 등 다양한 모델로도 변환이 가능하다. UML은 UML 메타모델이 지원이 되기 때문에 메타모델을 만들지 않아도 되지만 메타모델이 없는 모델일 경우 MOF(Meta Object Facility)[6]를 이용하여 메타모델로 설계가 요구된다.

모델 변환[7]을 수행하기 위해서는 소스 모델, 소스 메타모델, 타겟 메타모델, 변형 정의가 필요하다. 모델 변환 엔진은 이러한 요소들을 사용해서 타겟 모델로 변환한다. 모델 변환에서 중요한 요소로는 변환 룰을 정의하는 언어에 있다. UMT(UML Model Transformation)[8], MTL(Model Transformation Language)[9], QVT(Query / View / Transformation)[10], ATL(ATLAS Transformation Language)[11] 등이 있다. UMT는 XML, XMI, XSLT를 이용한 방법이다. 모델변환으로 입력되는 값이 XMI이지만 이는 근본적으로 XML이기 때문에 XML의 변환언어인 XSLT를 사용하여 변환한다. MTL은 DSL(Domain Specific Language)로 작성된 소스모델에 대해서 모델변환이 가능하며 자바가상머신의 개념을 응용해서 메타모델 언어로 작성된 메타모델을 일종의 컴파일러처럼 변환한다. QVT는 OMG에서 표준으로 추진하는 모델 변환 언어로 When, Where 등과 같이 구체적으로 변환된 시점을 정확하게 표현할 수 있는 장점이 있으면 패턴을 만들어 재사용할 수 있다. ATL은 MTL과 마찬가지로 변환정의언어 이면서 변환엔진으로 MDA 프레임워크를 이용한 모델변환을 지원한다. ATL은 무엇보다도 MTL이 가지고 있었던 변환규칙 정의가 보다 명확하고 이해하기 쉬우며 복잡한 변환규칙에 대한 정의가 가능하다. 또한 변환모델에 대해서 재사용 및 합성을 지원하는 장점을 가지고 있다. 본 논문에서는 여러 가지

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업(NIPA-2010-(C1090-0903-0004))과 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2010-0012117).

변환언어들 중에서 ATL의 이러한 장점 때문에 모델 변환 언어로 선택하였다

스마트폰의 하드웨어는 3.5 인치의 LCD와 터치센서, GPS, 중력, 가속도 등 다양한 센서들로 구성된다. 이것은 기존의 모바일 시스템과는 다르게 작은 화면으로 되어있는 스마트폰의 입력장치를 다양한 센서들로 보조하는 방식이다 또한 다양한 센서들 덕분에 악기, 3D 나침반, 내비게이션 등 다양한 콘텐츠들로 구성이 가능하고 개발 범위는 무궁무진하다 스마트폰의 소프트웨어 구성은 하드웨어 처리나 로직을 처리하는 소스와 화면의 UI나 그림파일등의 정보를 가지고 있는 리소스로 구성된다 대부분의 플랫폼들이 빠른 화면의 UI구성을 위해서 XML기반의 UI 편집 언어를 가지고 있다 이러한 형식의 소프트웨어는 MVC 아키텍처로 만들어 낼 수 있다. 대부분의 스마트폰의 소프트웨어가 이를 적용한 것은 아니지만 Apple의 iPhone은 이를 철저하게 지킨다. MVC는 소프트웨어를 모델 뷰, 컨트롤부분으로 분리하여 개발하는 방식으로 유지보수 및 이식성을 높일 수 있다

윈도우 모바일 플랫폼도 MVC를 지원하지 않지만 UI와 프로그램 코드를 분리되어있다 그림 3의 윈도우 모바일의 어플리케이션의 구조를 살펴보면 UI를 생성하는 Form.Desiner.cs와 이를 제어하는 Form.cs로 구분되어 있는 것을 확인가능하다 Visual Studio가 제공하는 UI 편집 도구를 사용하여 Button, Panel, Edit Text 등을 만들면 Form.Desiner.cs에 자동으로 UI 코드가 생성된다. 각 UI 컴포넌트가 발생될 이벤트를 처리해주는 핸들을 등록하고 Form.cs에서 구체적인 코드를 작성하는 형식으로 프로그래밍 한다. 이러한 어플리케이션의 구조를 통해서 타겟 독립 모델을View와 Controller의 형태로 구분하여 설계를 해야 한다는 것을 알아냈다 그래서 본 논문에서는 적용사례의 타겟 독립 모델을 설계에서 View와 Controller를 분리하였다.

3. 결론

본 논문에서는 이종의 스마트폰 개발을 위해서 윈도우 모바일 플랫폼 개발에 모델 변환 기법을 적용하였다. 모델 변환 기법을 적용하기 위해서 윈도우 모바일 플랫폼의 어플리케이션의 구조를 분석하였다 분석한 결과를 통해서 독립적인 특성과 종속적인 특성들로 구분하고 이를 기반으로 모델 변환 규칙을 만들었다. 만들어진 규칙은 5개의 규칙이며 크게 구분하면 기본구조클래스 단위, 메서드 단위 3가지로 분류화 할 수 있다. 또한 모델 변환 규칙을 ATL을 사용하여 기술하였고 이것을 이클립스 환경에서 모델 변환을 수행하였다 그 결과 모델 변환이 변환 규칙대로 클래스 다이어그램을 생성한 것을 확인 할 수 있었다. 윈도우 모바일 환경의 적용사례를 통해서 생성된 타겟 독립 모델이 이종의 다른 플랫폼으로 변환될 수 있는 가능성을 보았다 본 논문의 플랫폼 독립모델을 사용하고 모델 변환 규칙을 재정의 하면 아이폰안드로이드 등의 이종의 플랫폼으로 변환이 가능하다 향후연구로 아이폰 안드로이드 플랫폼을 분석하고 본 논문에 적용한 방법을 각 플랫폼에 맞게 확장할 것이다

참고문헌

- [1] Axel Jantsch, (2004), Modeling Embedded System and SOCs, Mogan Kaufmann
- [2] Woo Yeol Kim, Hyun S. Son, Young B. Park, Byung H. Park, C. R. Carlson, R. Young Chul Kim, "The Automatic MDA (Model Driven Architecture) Transformations for Heterogeneous Embedded Systems", Proceedings of The 2008 International Conference on Software Engineering Research and Practice, Vol. 2, 409-414, 2008.07.14.
- [3] Woo Yeol Kim, R. Young Chul Kim, "A Study on Modeling Heterogeneous Embedded S/W Components based on Model Driven Architecture with Extended xUML", The KIPS Transactions, Vol. 14-D, No. 1, 2007.02.
- [4] 손현승, 김우열, 서채연, 김동호, 김동우, 김재수, 김영철, "이종 임베디드 소프트웨어를 위한 코드 생성 메커니즘 및 지원도구", Proceedings of 2007 Korean Conference on Software Engineering, Vol. 9, No. 1, 170-177, 2007.02.22.
- [5] Woo Yeol Kim, Hyun Seung Son, R. Young Chul Kim, C. R. Carlson, "MDD based CASE Tool for Modeling Heterogeneous Multi-Jointed Robots", Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering, Vol. 7, 775-779, 2009.04.01.
- [6] OMG, Meta Object Facility Specification, In OMG Unified Modeling Language Specification, Version 2.0, January 2006.
- [7] K. Czarnecki, S. Helsen, "Feature-Based Survey of Model Transformation Approaches. IBM Systems Journal", Vol. 45, No. 3, pp. 621-64, 2006.
- [8] Roy Grønmo and Jon Oldevik, "An Empirical Study of the UML Model Transformation Tool(UMT)", In The First International Conference on Interoperability of Enterprise Software and Applications (INTEROP-ESA), Geneva, Switzerland, February 2005.
- [9] D. Vojtisek and J.-M. Je'ze'quel, "MTL and Umlaut NG: Engine and Framework for Model Transformation", http://www.ercim.org/publication/Ercim_News/enw58/vojtisek.html.
- [10] OMG, Documents associated with Meta Object Facility (MOF) 2.0 Query/View/Transformation, Version 1.0, April 2008.
- [11] J. Be'zivin, G. Dupe', F. Jouault, G. Pitette, and J. E. Rougui, "First Experiments with the ATL Model Transformation Language: Transforming XSLT into XQuery", Proceedings of the Workshop on Generative Techniques in the Context of Model Driven Architecture, Anaheim, CA, 2003.