Design by Contract 기법과 시큐어 코딩을 통한 안전한 소프트웨어 개발 연구

오준석*, 황대연, 최진영
고려대학교 컴퓨터학과
e-mail : {jsoh, dyhwang, choi}@formal.korea.ac.kr

Research Developing safety software with Design by Contract Method and Secure Coding

Joon-Seok Oh*, Dae Yon Hwang, Jin-Young Choi
Dept of Computer Science, Korea University

요 약

소프트웨어의 크기가 대형화되고 복잡화됨에 따라 소프트웨어의 기능성 오류 및 취약점을 개발 후 테스팅에 의해 찾는 비용이 매우 커지고 있다. 또한 테스트에 의한 방법을 통해 내재된 모든 오류나 취약점을 찾는 것은 거의 불가능하다고 인식되고 있다. 이러한 이유로 소프트웨어 개발에서 오류 및 취약성을 제거하고자 하는 노력이 증대되고 있다. 본 논문에서는 오류를 줄이고자 하는 기법중 하나인 Design by Contract와 취약성을 줄이고자 하는 시큐어 코딩을 소개하고, 이 두 가지 기법을 접목하여 오류가 없는 안전한 소프트웨어를 개발하는 방법을 소개한다.

1. 서론

소프트웨어는 운행 중에 오류 없는 올바른 동작을 하도록 보장되어야하고 내재된 취약성 및 해킹에 대해 저항이 강해야 한다. 특히 안전·보안 필수시스템에서 더욱 강조되어야 한다. 이런 시스템들의 특징은 운행 중에 소프트웨어 오류로 인해 사고가 나면 인명손실 및 재산상의 피해를 유발하기 때문이다. 이를 예방하기 위해 많은 연구가 진행되고 있으며, 본 논문에서는 안전한 소프트웨어를 개발하기 위한 방법으로 Design by Contract (DbC) 기법과 시큐어 코딩을 병행하여 소프트웨어를 개발하는 방법을 제시한다. DbC 기법은 오류를 줄이는 기법이고, 시큐어 코딩은 취약성을 줄이고자 하는 기법이다.

본 논문의 구성은 다음과 같다. 2장에서는 Design by Contract (DbC) 기법과 시큐어 코딩의 간단한 소개를 한다. 3장에서는 DbC 기법과 시큐어 코딩을 적절하게 접목시켜 개발에 적용할 수 있는 방법을 제시하며 4장에서는 결론을 맺는다.

2. 관련연구

· Design by Contract (DbC)

DbC 기법[1]은 시스템 운행에 있어서 원하는 기능을 수행할 수 있도록 소프트웨어를 개발하기 위한 기법들 중의 하나이다. DbC 기법의 기본원칙은 명세에 있는 제약들을 코드로 구현하는 것이다. 이런 제약들에는 전조건(precondition), 후조건(postcondition), 불변조건(invariant)이 있으며, 어노테이션(annotation)을 사용하여 프로그램 코드로 구현할 수 있다. 이런 어노테이션들은 프로그래밍 언어의 지원 하에 사용할 수 있으며 구현된 코드가 명세를 만족함을 검증하기 위해서는 다양한 검증도구들을 사용한다[2].

DbC 기법은 정형기법과 Hoare Logic에 근원을 두고 있다[3]. 정형명세를 통해 Hoare Logic에 있는 제약들을 명세하고 그 제약들을 코드로 구현한다. 코드로 구현된 제약들은 정형검증을 통해 코드가 명세를 만족하는지 검증한다. 이런 과정을 통해 코드가 명세를 만족함을 보장할 수 있고 올바른 동작을 하는 시스템을 개발하는데 도움이 되므로 보다 많은 소프트웨어의 오류를 줄일 수 있다.

· 시큐어 코딩 (secure coding)

시큐어 코딩은 소프트웨어 개발에서 취약성 및 해킹으로부터 안전한 소프트웨어 시스템을 만들기 위한 기법이다. 모든 소프트웨어 개발에 적용될 수 있는 기법이며, 특히 보안필수 소프트웨어에서 공격자들에 의해 악용될 수 있는 코드를 피하기 위한 방법들을 지침하고 있다[4]. 그러므로 시큐어 코딩기법을 적용하여 소프트웨어를 개발하면, 코드의 취약성을 줄일 수 있으며 해킹으로 인한 피해를 예방하고 보다 안전한 소프트웨어를 개발할 수 있다.

특정 프로그래밍 언어 및 개발환경에 대한 시큐어 코딩 규칙들이 해외에서 활발한 연구가 진행되고 있으며 본 논문에서는 특정 프로그래밍 언어에 국한되지 않고, 소프트웨어 개발에 있어서 공통적으로 사용될 수 있는 시큐어 코딩의 원칙 및 관습들을 소개한다.

본 연구는 교육과학기술부/한국과학재단 우수연구센터 육성사업 (R11-2008-007-03002-0)과 한국소프트웨어진흥원의 SW공학 요소기술 개발과 전문인력 양성사업의 지원으로 수행되었음 .

다음은 시큐어 코딩의 몇 가지 원칙 및 관습들이다.

① 코드를 작고 간단하게 유지해라
② 코딩스타일 가이드를 따르자
③ 안전한 코딩규칙・가이드라인을 따르자
④ 입력유효성검사 구현방법
⑤ 안전한 메모리 캐시 관리방법

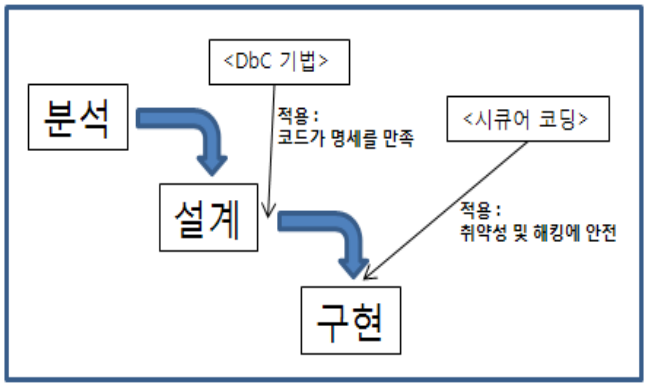
이 외에도 여러 원칙 및 관습들이 있으며, 보다 강건한 소프트웨어를 개발하기 위해서는 시큐어 코딩을 적용하여 개발하면 도움이 된다.

3. DbC 기법과 시큐어 코딩 접목연구

소프트웨어 개발단계 중 코드개발단계에서 DbC 기법과 시큐어 코딩을 적절하게 병행하여 코드개발에 적용하면 올바른 동작을 수행하면서 보다 강건한 소프트웨어를 개발하는데 도움이 되며 더불어 테스팅에 의해 소요되는 비용을 경감시킬 수 있다는 장점이 있다.

소프트웨어 개발에 있어서 시큐어 코딩을 적용하지 않고 DbC 기법만을 적용하면 소프트웨어가 안전하게 올바른 동작을 하도록 개발할 수 있지만, 개발된 코드가 잠재적인 취약성 및 해킹에 대해 안전하지 않은 코드구조를 가질 수 있는 단점이 있다. 또한 소프트웨어가 대형화되고 복잡해짐으로써 이런 오류들을 탐지하는데 소요되는 비용이 증대되고 있으며 테스팅을 통해 안전하지 않은 코드구조와 잠재적인 모든 취약성들을 탐지하는 것은 거의 불가능하다. 이와 반면에 DbC 기법을 적용하지 않고 시큐어 코딩만을 적용한다면 잠재적인 취약성에 대해 안전한 코드구조를 갖는 코드를 개발할 수 있지만, 대형화되고 복잡해진 시스템에 대하여 안전하게 올바른 동작을 수행하는지를 검증하는데 어렵다는 단점이 있다.

다음 [그림 1]은 소프트웨어 개발에 있어서 DbC 기법과 시큐어 코딩을 접목시킨 그림이다.



[그림 1] DbC 기법과 시큐어 코딩 접목

소프트웨어 개발에서 두 기법을 적절하게 병행하여 적용하면 각 기법들이 갖는 문제점들을 서로 보완할 수 있어 효과적이다. 이를 통해 보다 강건한 소프트웨어를 개발하는데 도움이 되며 오류 및 취약성을 찾는 테스팅에 대한 비용을 경감할 수 있기 때문에 경제적이다. DbC 기법을 분석, 설계로부터 만들어진 명세에 적용하면 올바른 동작을 하는 소프트웨어를 개발하는데 도움이 된다. DbC 기법을 사용하여 명세에 있는 제약들을 코드로 구현할 때, 여기에 시큐어 코딩을 적용하여 안전한 코드구조, 데이터 구조 등을 사용하여 개발한다. 이를 통해 예상치 못한 동작 및 잠재적인 취약성에 대해 더욱 안전한 시스템을 개발할 수 있으며 테스팅으로 인해 소요되는 비용을 경감할 수 있다.

개발자는 이 두 기법들을 적절하게 병행하여 소프트웨어 코드개발에 적용하면 DbC 기법에서 잠재적인 취약성 및 안전하지 않은 코드구조가 없음을 보장하지 못하는 단점을 시큐어 코딩을 통해 보완하고, 시큐어 코딩에서는 코드가 명세를 만족하여 시스템이 올바른 동작을 하도록 검증하는 것에 대해 어렵다는 단점을 DbC 기법을 통해 보완하였다.

5. 결론 및 향후연구

본 논문에서는 올바른 동작을 하고 잠재적인 취약성으로부터 안전한 소프트웨어를 개발하기 위한 방법으로 시큐어 코딩과 DbC 기법을 병행하여 적용하는 방법을 제시하였다. 소프트웨어 개발에 있어서 DbC 기법과 시큐어 코딩을 적절하게 병행하여 적용함으로써 각 기법들이 갖는 단점을 보완했으며 테스팅을 통한 시스템의 오류 및 내재된 취약점들을 발견하는데 소요되는 비용을 절감할 수 있도록 도움이 되었다.

향후연구는 안전한 소프트웨어를 개발하기 위한 방법론인 CbyC (Correctness by Construction)[6]에 본 논문에서 제시한 연구를 적용하여 실제 소프트웨어 개발에 적용하고자 한다.

참고문헌

[1] Meyer, B., "Applying "design by contract"". IEEE Computer, vol. 25, no 10, pp 40-51, Octal 1992.

[2] John Barnes, "High Integrity Software : The SPARK Approach to Safety and Security", Addison Wesley, 2002.

[3] D. Crocker. "Safe object-oriented software: The verified design-by-contract paradigm". In Redmill and Anderson, Editors, Proc. Twelfth Safety-Critical Systems Symposium, Springer (2004), pp. 19 - 41.

[4] M. G. Graff and K. R. Van Wyk. "Secure Coding : Principles and Practices". O'Reilly, 2003.

[5] Parnas, D. L.,van Schouwen, A.J., and Kwan, S.P. "Evaluation of safety-critical software". Commun. ACM 33, 6(JUN 1990), 636-648

[6] Roderick C. "Correctness by Contruction : a manifesto for high integrity software", Proceedings of the 10th Australian workshop on Safety critical systems and software, 2006