

AdaDoc-X: Ada 프로그램을 위한 자동 문서 생성 시스템의 구현¹⁾

김연정^o 이민수

이화여자대학교 과학기술대학원 컴퓨터학과
inverno@ewhain.net, mlee@ewha.ac.kr

Implementation of automatic document creating system: AdaDoc-X

Yearn Jeong Kim^o Minsoo Lee

Dept. of Computer Science&Engineering, Ewha Institute of Science and Technology, Ewha Womans University

요 약

국제적으로 표준화 된 객체지향 언어 중 하나인 ada는 동시수행제어(concurrent) 프로그래밍 언어로 임베디드 시스템에서 가장 널리 사용되고 있다. 하지만 현재 ada 프로그램의 주석을 처리하거나 프로그램의 명세를 나타내는 틀은 거의 없다. 이에 기존에 존재하는 javadoc과 비슷한 역할을 하는 AdaDoc을 단순히 주석들만 출력하는 것이 아닌 몇 가지 특정 tag를 지원하게 하고 명세들을 보기 좋게 출력할 수 있는 자동 문서 생성 시스템으로 확장하였다.

1. 서 론

Ada[1]는 국제적으로 표준화된 객체지향 언어로서, 본래는 군사용 실시간 프로그램 개발을 위해 만들어진 언어이나, 일반 응용 프로그램 개발에도 적합성이 입증된 범용 프로그래밍 언어이다. 어떤 상황에서도 사용할 수 있는 ada는 동시수행제어 프로그래밍 언어로 임베디드 시스템에서 가장 널리 사용되고 있다. Ada가 쓰이는 분야는 항공기, 항공관제 시스템, 회계 시스템, 통신 시스템, 의료기기 등의 다양한 분야에서 사용된다.

Ada 프로그램은 수많은 기업에서 그들 제품의 다양한 요구사항에 맞춰 작성되면 특히 오늘날 보안이나 안정성이 매우 중요해지면서 작성된 ada 프로그램이 얼마나 이러한 요구사항 및 신뢰성을 만족하는지를 분석하는 기술이 핵심적인 문제가 되고 있다. 대부분의 경우에 ada 프로그램을 작성할 때 다양한 요구(규칙, 조건)등에 대한 명세를 위해 각 모듈별로 주석을 추가하게 되는데 이들을 적절하게 추출하여 분석하기에 좋은 포맷으로 변환함으로써 다양한 분석을 가능하게 할 수 있다. 따라서 이러한 모듈 내부의 주석과 같은 정보를 파싱을 통해 다양한 포맷으로 출력할 수 있는 도구가 필요하다.

이러한 시도는 아직 국내에서는 시도된 적이 없으며, 국외에서는 AdaDoc이라는 도구를 통해 일부 기능을 제공하고 있다. AdaDoc은 Ada95 표준 프로그래밍 언어를 사용하는 개발자를 위한 도구로서, Ada 프로그램으로부터 다양한 포맷의 문서를 생성하는 도구이다. 현재, HTML, StrongHelp, Texinfo 그리고 Latex 포맷을 지원하고 있으며, 앞으로 더 많은 포맷을 지원할 예정이다. 아울러, AdaDoc은 GPL (GNU Public License) 라이선스를 갖고 있어, 소프트웨어를 자유롭게 공유하고 내용을 수정할 수 있다. 하지만 이 도구

는 주석에 관하여는 다른 처리 없이 그대로 출력하기 때문에 복잡한 요구사항을 처리하기에는 다소 무리가 있다.

본 논문에서는 AdaDoc을 확장하여 단순한 주석의 출력이 아닌, 복잡한 수식 및 조건, 규칙도 지원할 수 있는 확장된 Ada 포맷 출력 도구 구현에 대해 기술하고 있다.

2. AdaDoc

Adadoc[2]은 University of Applied Sciences of Western Switzerland의 학생이 만든 것으로 현재 출력 포맷으로는 HTML, LaTeX[3], StrongHelp, Texinfo를 지원한다. 현재 default값으로 html이 지정되어 있다.

AdaDoc의 구조를 살펴보면 입력으로는 오직 ada의 specification만 정의한 .ads파일만 받을 수 있고 ada의 body 파일인 .adb 파일은 지원하지 않는다.

AdaDoc의 parser를 거친 이 .ads 입력 파일은 임시적으로 xml 파일을 만들고 출력하고자 하는 출력 포맷의 모듈에서 이 임시 xml 파일을 읽어와 원하는 포맷으로 그대로 출력만 해준다.

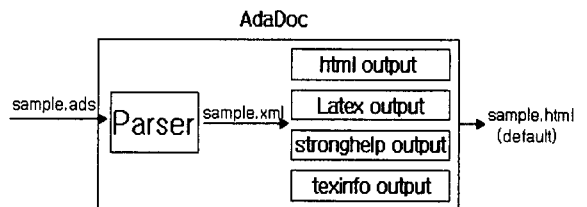


그림 1 AdaDoc의 구조

1) 이 연구과제는 정보통신 선도기반기술개발사업 A1100-0401-0143 를 수행중인 (재)그래픽스연구원의 위탁연구과제로 수행되었음

AdaDoc에서 기본적으로 지원하는 주석은 2가지 방식으로 분류가 되는데 선언 전에 위치하는 헤더 부분의 주석과 선언과 관계되는 주석이다. 헤더 부분의 주석은 일정한 키워드를 가진 태그로 구성되는데 기본적으로 AdaDoc에서 지원하는 태그는 다음과 같다.

표 1 AdaDoc에서 지원하는 헤더 태그들

헤더 부분 태그
@author
@date
@brief
@version
@revision
@filename

이 태그들 외에도 사용자가 필요로 하는 태그들을 추가할 수 있는 adadoc_tags.cfg 라는 파일을 두었다. 하지만 이 태그들은 오직 헤더에 위치해야 하고 procedure, function 등 선언과 관련된 주석들에 관한 세부 태그를 지원하지 않은 채 comment라는 태그에 해당 주석들이 모두 저장되어서 처리된다. 주석들을 XML로 변환한 파일을 보면 다음과 같다.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Package name="Module" filename="module.ads"
  brief="AdaDoc root module (class)" date="17.6.02"
  author="Julien Burdy & Vincent Decorges"
  file="module"
  xsi:noNamespaceSchemaLocation="AdaDocV2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

그림 2 헤더 부분 주석의 XML 변환

```
- <Procedure name="Initialize" comment="Purpose :
Initialize the module with options.¶¶ Self : Object
itself.¶ Options : The options string for the
module.¶">
- <Parameters>
  <Parameter name="Self" type="Object" mode="in
  out" />
  <Parameter name="Options" type="String"
  mode="in" />
</Parameters>
</Procedure>
```

그림 3 선언과 관계되는 주석의 XML 변환

그림 3에서 보는 바와 같이 주석을 변환시킨 중간 임시 파일인 xml 파일에는 주석부분을 comment라는 attribute 1개로 받아오기 때문에 따로 주석 내용에 대한 처리 및 규칙 설정 등을 할 수 없다. 아울러, 변수에 관련된 키워드 없이 그대로 입력하는 형태이기 때문에 비효율적이며, 직관적인 인터페이스를 제공해주지 못한다. 출력 포맷이 LaTeX인 경우를 보면 헤더 태그 관련된 주석에 대한 정보는 하나도 출력을 하지 않고 있으며 밑에 clause 절, procedure, function 에 관련된 주석들을

그냥 순서대로 출력만 해 주고 있을 뿐 정리된 상태로 출력하거나 출력 포맷을 어떤 형식에 맞추어서 출력하지는 못한다.

```
Clause | Comment
Diet_TDA |
Clause | Comment
Diet_TDA |
Name
Object
Object_Access
Element_Type
Example : Use for know when an element start.
```

Initialize (procedure)

Purpose : Initialize the module with options.

Self : Object itself.

Options : The options string for the module.

Name	Type	Mode	default	Comment
Self	Object	in out		
Options	String	in		

그림 4 LaTeX 출력 결과

기존 AdaDoc의 이러한 부족한 점들을 보완하고 새로운 기능들을 추가, 확장한 AdaDoc-X을 구축하였다.

3. AdaDoc의 확장 - AdaDoc-X

3.1. 세분화된 태그 지원

기존 AdaDoc이 헤더 태그만을 지원하고 선언과 관련된 주석들에 관한 태그들이 없기에 태그들을 만들어서 추가하였다. 추가된 태그들은 표2와 같다.

표 2 AdaDoc-X에서 지원하는 태그들

Compilation Unit Preamble	
@filename	파일의 이름과 관련 주석들
@author	파일을 쓴 사람의 이름
@date	마지막으로 수정한 날짜(DD-MM-YYYY)
@brief	파일에 대한 간단한 설명(한 문장)
@file	파일에 대한 약간 자세한 설명
Package Documentation	
@package	패키지에 대한 설명
@brief	패키지에 대한 간단한 설명(한 문장)
Clause Documentation	
@clause	clause절에 앞에 써야 하며 패키지와의 연관성/의존도를 설명
Declaration Documentation	
@var	변수의 목적을 간단히 설명
@const	상수의 목적을 간단히 설명
@type	타입과 서브타입의 목적을 간단히 설명

Subprogram Documentation-Procedure	
@proc	프로시저에 대한 설명
@brief	프로시저에 대한 간단한 설명
@pre	프로시저의 선조건을 명시(없는 경우는 none이나 unspecified)
@inv	프로시저를 위한 하나 이상의 invariant
@post	프로시저의 후조건을 명시(없는 경우에는 none이나 unspecified)
@param	파라미터에 대해 간단히 설명
@note	프로시저와 관련해서 추가적인 정보를 기입하고 싶을 때 사용(선택)
Subprogram Documentation-Function	
@func	함수에 대한 설명
@brief	함수에 대한 간단한 설명
@pre	함수의 선조건을 명시(없는 경우에는 none이나 unspecified)
@inv	함수를 위한 하나 이상의 invariant
@post	함수의 후조건을 명시(없는 경우에는 none이나 unspecified)
@param	파라미터에 대한 간단한 설명
@return	리턴값에 대한 간단한 설명
@note	함수와 관련해서 추가적인 정보를 기입하고 싶을 때 사용(선택)

3.2. XML 트리 구조 변경

기존 AdaDoc에는 태그가 헤더에 관한 것 하나만 지원하므로 XML도 그림 2에서와 같이 package 태그로 시작해서 attribute로 헤더 태그만 넣었다. 하지만 여러 태그를 지원하게 되면서 위와 같은 구조의 XML은 변경시킬 필요가 있기에 XML의 최상위 태그로 compilation unit을 추가하고 그림 5와 같이 .ads 파일 내용을 순서대로 XML 태그로 만드는 형식으로 기존 XML 트리 구조를 그림 6과 같이 변경시켰다.

```
-- @file elaborate description of this file
-- @filename sample.ads
-- @author Yeam Jeong Kim
-- @date 4.29.2005
-- @brief sample code to test AdaDoc-X
-- @clause use Text_IO to print
with Text_IO;
-- @package package name is sample
-- @brief testing
package Sample is
```

그림 5 sample.ads

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <CompilationUnit file="elaborate description of this file"
  filename="sample.ads" brief="sample code to test AdaDoc-X"
  date="4.29.2005" author="Yeam Jeong Kim" name="sample"
  xsi:noNamespaceSchemaLocation="AdaDocV2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <Clauses>
- <ContextClauses>
  <Clause name="Text_IO" clause="use Text_IO to print" />
</ContextClauses>
</Clauses>
- <Package name="Sample" brief="testing" package="package name is
  sample" />
```

그림 6 sample.xml

3.3. 분류된 LaTeX 출력

AdaDoc-X에서는 출력 포맷은 LaTeX에 초점을 맞추었다. 기존 AdaDoc에서 헤더 태그를 전혀 출력해 주지 않기 때문에 변경된 헤더 태그 및 추가된 태그들을 LaTeX 출력물에 반영하였고 정리되지 않은 채 .ads 파일의 순서대로만 출력하던 기존 방식을 바꾸어서 예를 들면 procedure를 그룹핑해서 같이 출력한 후 function들을 출력하는 등 일정한 형식에 맞추어 출력하였다. 출력 형식은 다음 그림 7과 같다.

1 Compilation Unit Preamble

```
filename sample.ads
author Yeam Jeong Kim
date 4.29.2005
brief sample code to test AdaDoc-X
file elaborate description of this file
```

2 Clause Documentation

```
2.1 with Text_IO
use Text_IO to print
name Text_IO
```

3 Package Documentation

```
package package name is sample
brief testing
name Sample
```

그림 7 LaTeX 출력물

4. 결론 및 향후 계획

AdaDoc-X를 구현함으로써 임베디드 시스템에서 많이 사용하는 Ada 언어의 documentation을 기능을 많이 향상시킬 수 있으나 아직 확장자가 ads인 파일만 지원한다. 이 외에도 출력 포맷을 LaTeX에만 중점을 두었으므로 다른 포맷의 추가 및 확장이 필요하다.

5. 참고문헌

- [1] 신형강, 이남용, 'Ada 언어와 응용 : Ada 에 의한 소프트웨어 공학', 1987
- [2] AdaDoc, <http://adadoc.sourceforge.net/>
- [3] 신병춘, 최근배, '아름다움을 만날 수 있는 문서 작성 시스템 LaTeX 2ε 길라잡이', 2004