KOROSKOR
OROSKOR
ROSKOR
OSKOR
SKOR
KOR
OR

KOROS

서비스 로봇을위한 모듈 - 제2-2부 : 소프트웨어 모듈을 위한 정보 모델

KOROS XXXX : YYYY

한국로봇산업협회

YYYY년 MM월 DD일 제정

http://www.KOROS.or.kr

KOROS XXXX : YYYY

심 의 : 지능형로봇표준포럼 운영위원회

	성 명	근 무 처	직 위
(운영위원장)	문 승 빈	세종대학교	교 수
(운 영 위 원)	이 순 걸	경희대학교	교 수
	우 종 운	한국로봇산업진흥원	센 터 장
	곽 관 웅	세종대학교	교 수
	조 영 조	한국전자통신연구원	책임연구원
	김 동 한	경희대학교	교 수
	정 영 숙	한국전자통신연구원	책임연구원
	전 진 우	한국로봇산업진흥원	실 장
	임 성 수	경희대학교	교 수
	문 인 혁	동의대학교	교 수
	권 용 관	대림대학교	교 수
	김 규 로	국가기술표준원	국 장
	김 석 중	하이젠모터	이 사
	김 승 훈	전자부품연구원	책 임
	김 혜 지	국가기술표준원	주 무 관
	단 병 주	LG전자	수석연구원
	류 영 선	라스테크	대 표
	문 전 일	한국로봇산업진흥원	원 장
	박 종 환	널판	대 표
	박 홍 성	강원대학교	교 수
	서 준 호	한국로봇산업협회	팀 장
	서 태 원	로보테크	부 장
	심 재 홍	한국산업기술대학교	교 수
	유 원 필	한국전자통신연구원	책임연구원
	이 병 우	한국산업기술시험원	전문위원
	이 용 국	현대로보틱스	수석연구원
	임 우 철	두산로보틱스	선 임
	장 민	유진로봇	이 사
	정 병 찬	한화정밀기계	수석연구원
	지 수 영	한국전자통신연구원	책임연구원
(간 사)	유 기 은	한국로봇산업협회	대 리

표준작성 기여자

	성 명	근 무 처	직 위
(과제제안자)	O O O	OOOOOOOOOOOOO	O
(작성참여자)	O O O	OOOOOOOOOOOOO	O
	O O O	OOOOOOOOOOOOO	O
	O O O	OOOOOOOOOOOOO	O
	O O O	OOOOOOOOOOOOO	O
	O O O	OOOOOOOOOOOOO	O

목 차

서 문

표준의 목적

본 표준은 서비스 로봇에 사용되는 소프트웨어 모듈의 공통 특성을 나타내는 소프트웨어 모듈의 정보 모델을 제시한다. 소프트웨어 모듈의 정보 모델은 다양한 유형의 소프트웨어 제조업체 및 설계자가 독립적으로 개발 및 생산한 로봇 소프트웨어 모듈의 통합을 용이하게 한다. 서비스 로봇을 위한 모듈 – 제2-1부에 명시된 CIM 항목의 메소드와 속성을 상속받으며, 장치연동 소프트웨어 모듈과 순수소프트웨어 모듈로 구분하여 필수적 기본기능을 위한 함수 및 속성을 클래스 다이어그램을 통하여 제시한다.

.

타 표준과의 관련성

타 국내표준과의 관련성

KOROS 1067-3:2018 개방형 로봇 소프트웨어 플랫폼 제3부: 프로파일

KOROS 1149-1:2020 서비스 로봇을 위한 모듈 – 제1부 : 일반요구사항

KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부 : 모듈을 위한 공통 정보 모델

타 국제표준과의 관련성

ISO 22166-1 Modularity for service robots – Part 1 : General requirements

지적재산권 인지 관련사항

해당사항 없음

적합인증 관련사항

해당사항 없음

표준의 이력

본 표준은 YYYY년 MM월 DD일에 개최된 제OO차 지능형로봇표준포럼 운영위원회에서 제정되었다.
본 표준은 YYYY년 MM월 DD일에 개최된 제OO차 지능형로봇표준포럼 운영위원회에서 KOROS XXXX:YYYY를 개정하였다.

KOROS XXXX : YYYY

서비스 로봇을 위한 모듈 – 제2-2부 : 소프트웨어 모듈을 위한 정보 모델

Modularity for service robots – Part 2-2 : Information model for Software Modules

1 적용범위

이 표준은 서비스 로봇에 사용되는 소프트웨어 모듈을 대상으로 한다. 소프트웨어를 좀 더 상세하게 분류하면 장치 액세스 모듈과 순수 소프트웨어 모듈의 두 그룹으로 분류하는데, 전자는 로봇 제조업체에 공급할 수있는 소프트웨어 모듈개발자, 후자는 제조업체 / SW모듈 및 로봇통합자, SW 설계자 / 디자이너를 대상으로한다. 이 표준은 로봇 도메인에서 통합 및 재사용을 가능하게하는 일종의 도구로 간주되며 특정 구현 또는 서비스 로봇 유형에 관계없이 모듈 간의 상호 운용성 및 호환성을 촉진한다.

2. 인용표준

다음 문서는 전체 또는 일부가 이 문서에서 규범적으로 참조되며 해당 응용 프로그램에 반드시 필요하다. 날짜가 기입 된 참고 문헌에 대해서는 인용 된 판만 적용된다. 날짜가 명시되지 않은 참조의 경우, 참조 문서의 최신 버전 (모든 개정 포함)이 적용된다.

KOROS 1149-1:2020 서비스 로봇을 위한 모듈 – 제1부 : 일반요구사항
KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부 : 모듈을 위한 공통 정보 모델
IEEE/Open Group 1003.1-2017 IEEE Standard for Information Technology--Portable Operating System Interface (POSIX(TM)) Base Specifications, Issue 7

3. 용어와 정의

이 표준의 목적을 위하여 다음의 용어와 정의를 적용한다.

3.1
정보모델 (Information model)
IM
동작 환경, 속성 및 함수 그리고 서로 간의 관계를 가진 객체의 추상화와 표현

3.2
공통 정보 모델 (Common Information model)
CIM
모든 모듈에 공통으로 적용되는 정보 모델

3.3
모듈 (module)
시스템 디자인과 통합 그리고 상호운용성 및 재사용성을 도모하기 위하여 속성 프로파일에 수반되어 정의된 인터페이스로 복합 또는 결합한 컴포넌트

3.4
속성 (property)
모듈의 특성 또는 변수

3.5
소프트웨어 모듈 (software module)
SW 모듈
프로그램된 알고리즘으로만 구성된 모듈

3.6
인스턴스 (Instance)
동작하거나 사용되는 프로세스 혹은 객체

3.7
서비스 (service)
제공되는 인터페이스를 통하여 다른 모듈을 위한 하나 이상의 기능 또는 동작

3.8
미들웨어 (middleware)
모듈형 응용을 위해 데이터를 송수신하고 관리하는 소프트웨어.

비고 미들웨어은 로봇 미들웨어를 말하며, 그 예로는 ROS, openRTM, OPRoS 등이 있음

4. 소프트웨어 모듈의 정보 모델

4.1 개요

소프트웨어 모듈의 정보 모델 (SIM)은 KOROS 1148:2021 을 기반으로 해야 한다. 그림 4.1의 소프트웨어 모듈의 정보모델은 공통정보모델에서 상속받는다. SIM은 그림 4.2와 같이 CIM을 상속한다

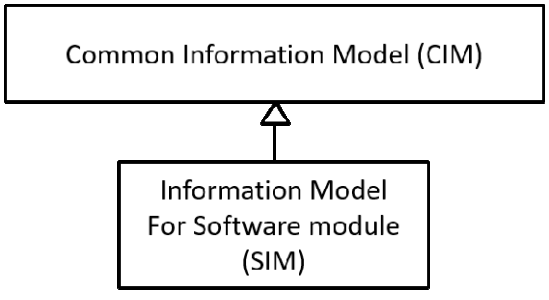

그림 4.1 — SIM 과 CIM의 상관도

비고 1 소프트웨어 모듈의 정보 모델은 소프트웨어 모듈의 디자인 및 배포 그리고 운영에 필요한 정보를 제공한다.

소프트웨어 모듈을 위한 정보모델 표 4.1의 'M' 과 'O'의 기호는 필수항목과 선택항목을 의미한다.

표 4. 1 — 소프트웨어 정보 모델과 공통 정보 모델의 대응 태그

번호.	속성 종류	필수(M) or 선택(O) 소프트웨어 모듈	관련 그룹/태그 이름 (각 그룹의 약자)
1	모듈이름 (Module Name)	M	GenInfo
2	설명 (Description)	O	
3	제조자 (Manufacturer)	M	
4	예제 (Examples)	O	
5	모듈아이디 (Module ID)	M	IdnType
6	소프트웨어 특성 (Software Aspects)	M	
7	모듈속성 (Module properties)	M	ModuleProp
8	입력 (Inputs)	M	IOVariable
9	출력 (Outputs)	M	
10	기능성 (Function(capabilities))	M	Service
11	인프라 (Infrastructure)	M	Infra
12	안전/보안 (Safety/Security)	M	SafeSecure
13	모델링 (Modelling)	O	Modelling

표 4.1의 모듈 이름, 설명, 제조자, 예제, 모듈아이디, 소프트웨어 특성, 모듈 속성, 입력, 출력, 기능성, 인프라, 안전/보안, 모델링 항목들은 KOROS 1148:2021 표4.1의 속성종류 항목에서 하드웨어 특성을 제외한 항목이다.

비고 2 소프트웨어 모듈은 연산 모듈에서 사용되거나 또는 하드웨어 소프트웨어 복합특성 모듈인 HW-SW모듈에 사용된다.

4.2 소프트웨어 모듈을 위한 정보모델 클래스

4.2.1 개요

SIM의 클래스는 KOROS 1148:2021 그림 4.5.1의 CIM 공통정보모델 클래스를 상속받아 사용한다. 그림4.2의 IDnType, Propertiees, IOVariables, Status, Services, Infra, SafeSecure, Modelling 블럭은 공통정보모델을 상속받은 것이며 추가로 ExecuteFiles블록이 추어 소프트웨어 모듈의 실행파일정보를 제공한다.
소프트웨어 정보모델 클래스 그림 4.2는 소프트웨어모듈의 동작중에 필요한 정보모델의 정보들을 제공하기 위한 GetSIM 인터페이스를 제공하여 해당 정보에 접근할 수 있도록 한다. 그림4.3은 소프트웨어 모듈의 정보모델 클래스이며 그림 4.4는 소프트웨어 정보모델의 클래스 객체들을 반환하는 인터페이스들이다.

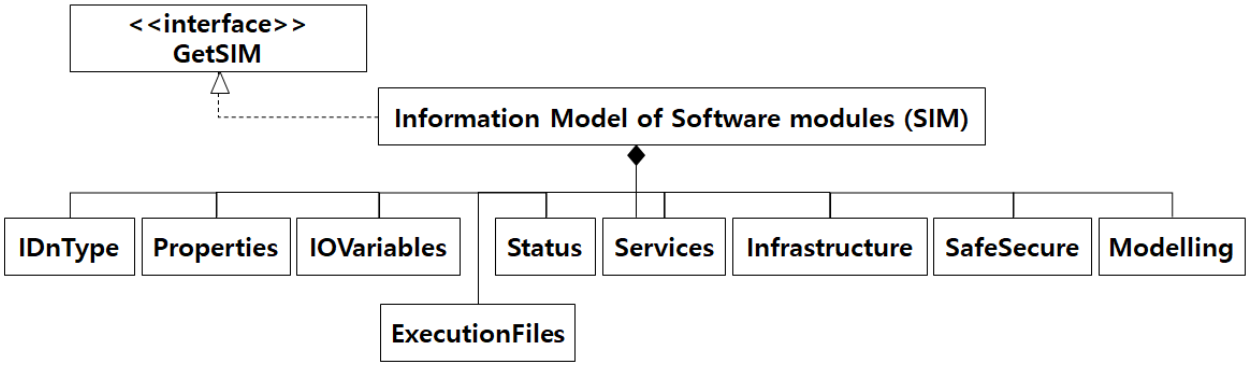

그림 4.2 — SIM을 구성하는 클래스와 인터페이스

비고 1 SIM 클래스는 KOROS 1149-1:2020에서 도출된 표 4.1의 해당 클래스인 IDnType, Properties, IOVariables, Services, Infra, SafeSecure, Modelling 블록들과 KOROS 1148:2021에서 추가된 클래스 Status 블럭을 가지고, 소프트웨어 모듈을 위한 클래스 ExecutionFiles를 추가한다.

비고 2 SIM을 구성하는 클래스는 표 4.1의 그룹이름을 가진 클래스들이며, 그중 ModuleProp은 Properties 클래스로 사용한다. CIM을 구성하는 클래스와 동일하다.

정보모델 버전은 모듈이 명시한 정보모델의 버전이다. 표준이 수정될때 마다 모델버전은 수정한다. 변수의 이름은 소문자 카멜표기법으로 한다.

비고 3 접근 제한자 private(-), protected(*), public(+) 은 KOROS 1148:2021의 그림 4.5 CIM클래스 다이어그램에 적용된 동일한 표식이다. 변수 이름 다음에 변수 타입을 명시하며, 변수이름과 변수타입의 구분자는 콜론을 사용하고, 만일 변수가 public으로 선언되면 해당 변수에 접근하기 위한 함수의 선언이 필요하지 않다.

Class SIM
- moduleName : String - description : String - manufacturer : String + examples : String -informationModelVersion : String -executionFile : ExecutionFiles - idnType : IDnType - properties : Properties - ioVariables : IOVariables - status : Status - services : Services - infra : Infra - safeSecure : SafeSecure -modelling : Modelling

그림 4.3 — SIM 클래스 다이어그램

<<Interface>> GetSIM
+ getModuleName() : String//모듈 이름 반환
+ getDescription() : String //모듈의 설명 반환
+ getManufacturer() : String //모듈의 제조자정보 반환
+ getInformationModelVersion() : String//모듈버전정보 반환
+ getExecutionFiles(): ExecutionFiles//모듈실행파일정보반환
+ getIdnType() : IDnType //모듈의 IDnType 객체 반환
+ getProperties() : Properties//모듈의 Property 객체 반환
+ getIoVariables() : IOVariables//모듈의 IOVariable객체 반환
+ getStatus() : Status//모듈의 Status 객체를 반환
+ getServices() : Services//모듈의 Service 객체를 반환
+ getInfra() : Infra//모듈의 Infra 객체를 반환
+ getSafeSecure() : SafeSecure//모듈의 SafeSecure 객체반환
+ getModelling() : Modelling//모듈의 Modelling 객체를 반환

그림 4.4 — SIM 클래스를 위한 인터페이스 GetSIM 다이어그램

비고 4 데이터 타입은 KOROS 1148:2021의 부속서 표C.4의 하나이어야 한다

4.2.2 소프트웨어 모듈의 모듈 아이디를 위한 클래스

소프트웨어 모듈의 아이디와 타입에 관한 정보는 그림 4.5과 같으며, KOROS 1148:2021의 '4.3.2 모듈 아이디와 모듈 유형 클래스'에 있는 IDnType클래스의 hwApsects 변수를 가지지 않는다.

보기 1 클래스 IDnType에 대한 정보와 SW특성이 있는 소프트웨어 모듈의 예는 다음과 같으며, 부속서 A.3에 명시방법을 제공한다.

```
<?xml version="1.0" encoding="UTF-8" ?>
< IDnType>      <!-- example of a composite module -->
  <ModuleID> moduleId </ModuleID>
  <InstanceID> instanceId </InstanceID>
  < SWAspects >
    (swModuleID1, swModuleID1InstnaceID), (swModuleID2, swModuleID2InstanceID)
  </ SWAspects >
  < IDnTypeService type = "IDL">
    Path of IDL file.
  </ IDnTypeService >
</ IDnType>
```

보기 2 IDL형식으로 정의한 IDnType의 함수의 정의는 다음과 같다.

```
interface IDnTypeService : CIMService
{
        uint8 getInstanceID();// 변수 instanceID 값을 반환하는 함수
}
```

단순 연산을 하는 소프트웨어 모듈은 소프트웨어특성 항목에 입력된 값이 없을 수 있다. 복합 소프트웨어 모듈은 소프트웨어특성 항목에 instanceID를 포함한 moduleID가 하나 이상의 입력을 가진다.

비고 동일한 모듈을 다른 서비스로봇의 응용에 재사용하여 구성하는 모듈통합자는 해당 모듈의 instanceID에 새로운 값을 할당한다.

Class IDnType
+ moduleID : String - instanceID : uint8 // 객체 ID + swAspects: Array of pair (moduleID, instanceID)
+ getInstanceID() : uint8 //모듈의 인스턴스 ID를 반환

그림 4.5 — CIM의 IDnType클래스를 확장한 IDnType클래스 다이어그램

4.2.3 소프트웨어 모듈의 실행파일을 위한 클래스

소프트웨어 모듈의 실행파일의 정보는 그림 4.6과 같으며, 실행파일의 이름과 실행에 필요한 라이브러리를 명시하여 모듈의 동작에 필요한 정보를 제공한다.

보기 그림 4.6의 클래스 ExecutionFiles에 대한 예는 다음과 같으며, 부속서 A.4에 명시방법을 제공한다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ExecutionFiles>
  <MainFileURL> moduleExecutionFileName </ MainFileURL >
  <LibraryURL> moduleRunTimeLibrary </ LibraryURL >
</ ExecutionFiles >
```

Class ExecutionFiles
+ mainFileURL: StringList //DLL/So/Exe of module + libraryURL : StringList // used in the module

그림 4.6 — SIM의 ExecutionFiles클래스 다이어그램

4.2.4 소프트웨어 모듈의 속성을 위한 클래스

소프트웨어 모듈의 속성에 관한 정보는 그림 4.7과 같으며, KOROS 1148:2021의 '4.3.3 모듈 속성을 위한 클래스'에 있는 Properties 클래스를 확장한다. 소프트웨어 모듈은 구동에 필요한 정보가 추가되는데, 프로그램밍 언어, 구동 컴파일러타입, CPU타입이 있다.

보기 그림 4.7의 예제는 다음과 같으며, 부속서 A.5에 명시방법을 제공한다.

```
<Properties languag="C++" osType="Ubuntu 20.04" compiler="g++", cpuTtype="x86-64" >
  <Property name="length", type="float32" value = "100.0" unit="millimetre" description =
"length of cylinder" />
  <Property name="radius", type="float32" value = "3.0" unit="millimetre" description =
"radius of cylinder" />
</Properties>
```

위의 태그로부터 다음과 같은 변수와 데이터타입을 생성할 수 있다,

+ length = 100.0: float32 or public float32 length =100.0

변수로 정의되어진 모듈의 속성값에 다른 모듈이 접근한 방법은 KOROS 1148:2021의 '4.3.3 모듈 속성을 위한 클래스'에 있는 접근 방법과 동일하다.

Class Properties
typedef Property[] PropertyList + property : PropertyList // Property 클래스에 따라 생성된 변수의 열겨형 + Language : Enumeration // 프로그래밍 언어 + OSType : Enumeration // 운영체제 type + CompilerType : Enumeration // 컴파일러 타입 + CPUType : Enumeration // 사용된 CPU 비트 수: 8, 16, 32, 64

그림 4.7 — CIM의 Properties 클래스를 확장한 Properties 클래스 다이어그램

4.2.5 소프트웨어 모듈의 입력 출력을 위한 클래스

소프트웨어 모듈의 입력 출력 공유변수에 관한 정보는 그림 4.8과 같으며, KOROS 1148:2021의 '4.3.4 모듈의 입력 출력 및 공유변수를 위한 클래스'에 있는 IOVariables 클래스를 확장한다. 소프트웨어 모듈은 Variable의 입력과 출력을 함수를 통해서 접근할 수 있도록 접근 제한자를 private으로 확장하여 함수를 제공한다.

모듈의 변수를 다른 모듈이 접근한 방법은 KOROS 1148:2021의 '4.3.3 모듈 속성을 위한 클래스'에 있는 접근 방법과 달리 추가된 setVariables()함수와 getVariables()함수를 통하여 교환한다.

보기 1 그림 4.8을 위한 예제는 다음과 같으며, 부속서 A.5에 명시방법을 제공한다.

```
<IOVariables>
   <Variable variavleName=”controlValue” ioType="IN" variableType=”float32” variableUnit=ampere variableDescription=“current control command to motor” > <value />
   </Variable>
  <Variable variableName=”encoder” ioType="OUT" variableType =”uint32” variableUnit = “none” variableDescription = “value of absolute encoder” variableValue= 0 />
  </Variable>
  <!-- input and output -->
  <Variable variableName=”state” ioType="INOUT" variableType=”uint8” variableUnit = “none” variableDescription = “ reset or read status of motor”>
        <value />
   </Variable>
   < IOVariableService type = “IDL”>
    Path of IDL file.
   </ IOVariableService >
</IOVariables>
```

위의 태그들로부터 다음과 같은 변수와 데이터 타입이 생성될 수 있다.

```
+ controlValue : float32   or public float32 controlValue // 현재의모터 제어 명령어
+ encoder=0 : float32      or public float32 encoder=0   // 앱솔루트 엔코더의 값
+ state : uint8            or public uint8 state         // 모터의 상태를 읽거나 재설정
```

보기 2 IDL형식으로 정의한 IOVariables의 함수의 정의는 다음과 같다.

```
interface IOVariableService : CIMService
{
        Boolean setVariable(Variable variable); // 제공된 variable 값을 추가하는 함수
        any getVariable(Variable variable); // 요청된 variable 값을 반환하는 함수
}
```

Class IOVarables
- ioVariables[] : Variable
+setVariable(variable: Variable):Boolean//subscriber,데이터입력포트,공유메모리WRTIE로사용 + getVariable(variable: Variable):any//publisher,데이터출력포트,공유메모리READ로 사용

그림 4.8 — CIM의 IOVarables를 확장한 클래스 다이어그램

4.2.6 소프트웨어 모듈의 상태를 위한 클래스

소프트웨어 모듈의 상태 관련 정보는 그림 4.9과 같으며, KOROS 1148:2021의 '4.3.5 모듈의 상태를 위한 클래스'에 있는 Status 클래스를 확장한다. 소프트웨어 모듈은 변수인 ExecutionStatus, ErrorType을 함수를 통해서 접근할 수 있도록 접근 제한자를 private으로 확장하여 함수를 제공한다.

비고 1 소프트웨어 모듈의 5개 상태는 함수에 의해서 전의를 하는데, 그림 4.9의 예시에서는 initialize함수가 모듈이 동작에 필요한 초기화를 수행하며, 모듈의 상태를 created에서 idle로 전환한다. start함수는 소프트웨어 모듈이 제공하는 서비스를 시작시키고, run함수와 stop함수가 모듈의 수행과 대기를 반복하며 executeing 상태와 idle상태를 반복한다. 모듈은 destroy 함수에 의해서 idle상태 또는 executing상태에서 소멸될 수 있다. 함수 error는 모든 상태에서 모듈의 상태는 error로 전환한다.

소프트웨어 모듈의 Status클래스는 모듈의 상태를 나타내는데 소프트웨어 정보모델이 존재하는지와 존재한다면 통신의 상태나 동작중 에러가 발생했는지를 알리는 실행주기의 상태가 명시되는지로 확인할 수 있다. ExecutionStatus는 KOROS 1149-1:2020의 그림6과 같은 아래의 그림4.9의 상태 (created, idle, executing, destructed, error) 중 하나 이어야 한다.

소프트웨어 모듈의 상태에 대한 데이터는 상위 표준인 KOROS 1149-1:2020의 7.3 소프트웨어 모듈 아키텍처 모델 의 그림5에 있는 보안 및 안전 관리자가 요청할시 모듈이 제공해야할 정보이다.

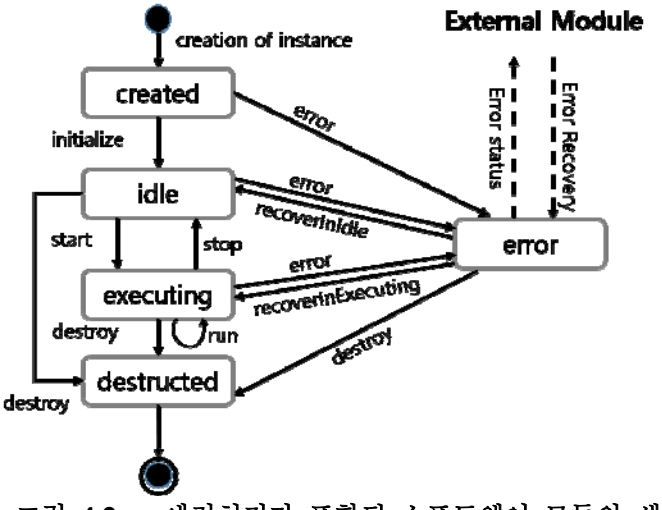

그림 4.9 — 에러처리가 포함된 소프트웨어 모듈의 생명주기

변수 들은 errorType과 같이 추후에 정의한다.

비고 2 에러 타입은 KOROS 1148:2021의 '4.3.5 모듈의 상태를 위한 클래스'에 사용된 에러 번호들처럼 IEEE/Open Group 1003.1-2017 IEEE Standard for Information Technology--Portable

Operating System Interface (POSIX(TM)) Base Specifications, Issue 7에 따르며, 소프트웨어 모듈을 위해서 추가되는 에러번호들은 POSIX.1003.1에 위반되지 않도록 정의해야한다.

비고 3 KOROS 1148:2021의 '4.3.5 모듈의 상태를 위한 클래스'에 명시된 errorType 예시는 Operation not permitted, Authentication Error, Bad Parameter, Unsupported Service, out of Ranges, Precondition not met 이다.

동일한 모듈ID의 다른 인스턴스ID를 갖는 모듈은 별개의 모듈이어서 다른 상태의 값을 가질 수 있다.

보기 1 그림 4.10을 위한 예제는 다음과 같으며, 부속서 A.6에 명시방법을 제공한다.

```
<Status >
    <ExecutionStatus exeStatus="created"/>
    <ErrorType errorType="OK"/>
    < StatusService type = "IDL">
     Path of IDL file.
    </ StatusService >
</Status>
```

보기 2 IDL형식으로 정의한 Status의 함수의 정의는 다음과 같다.

```
interface StatusService: CIMService
{
        ExeStatus getExecutionStatus ();//현재의 exeStatus값을반환하는함수
        putExecutionStatus (ExeStatus status);// 제공된 status 값으로 설정하는 함수
        int32 getErrorType ();// 현재의 errorType 값을 반환하는 함수
        putErrorType (int32 error);// 제공된 error 값으로을 설정하는 함수
}
```

Class Status
typedef ExeStatus enumeration of {CREATED, IDLE, EXECUTING, DESTRUCTED, ERROR} - ExecutionStatus : ExeStatus // - ErrorType : int32 // depending on module type, enumeration will be defined

+ getExecutionStatus() : ExeStatus // 상태정보를 반환하는 함수
+ putExecutionStatus(ExeStatus status) // 상태정보를 설정하는 함수
+ getErrorType() : int32 // 에러타입정보을 반환하는 함수
+ putErrorType(int32 error) // 에러타입정보를 설정하는 함수

그림 4.10 — CIM의 Status를 확장한 Status 클래스 다이어그램

안전 관리자는 구동할 응용을 구성하는 모듈들로 부터 에러관련 정보와 복구관련 정보를 제공받는다. 명시된 개별 안전수준과 보안수준을 SafeSecurity 클래스의 함수들을 호출하여 수집하고, 응용의 안전 레벨과 보안 레벨을 분석한다. 레벨을 구성하여 분석을 한다.

4.2.7 소프트웨어 모듈의 서비스를 위한 클래스

소프트웨어 모듈의 서비스를 위한 정보는 KOROS 1148:2021의 '4.3.6 모듈의 서비스를 위한 클래스'를 적용한다.

4.2.8 소프트웨어 모듈의 인프라를 위한 클래스

인프라 클래스의 속성 중 DataBus 클래스는 그림 4.11과 그림4.12와 같이 모듈이 제공하는 인프라

타입을 상위 표준인 KOROS 1148:2021의 인프라 클래스를 구성하는 DataBus클래스를 상속받아 확장한다.

소프트웨어 모듈이 하드웨어 장치로부터 데이터를 획득할때는 Databus에 명시된 통신방법을 사용한다.

비고 1 하드웨어 장치로 부터 데이터를 획득하기 위해서 인프라 모듈이 사용하는 함수는 '4.2.7 모듈의 서비스를 위한 클래스'를 사용하여 정의해야 한다.

비고 2 순수 소프트웨어 모듈의 정보모델은 DataBus클래스의 값을 갖지 않는다.

보기 1 그림 4.11을 위한 예제는 다음과 같으며, 부속서 A.8에 명시방법을 제공한다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Infra>
  <DataBuses>    <!-- list of data buses used in Module -->
   <DataBus>
      <ConnectorType> DE9 </ConnectorType>       <!-- D-Sub 9-->
      <TypePhyMac> CAN </Type>
      <TypeNetTrans> CANopen </TypeNetTrans>
      <TypeApp> OBD NMT SDO PDO SYNC </TypeApp>    <!-- CANopen -->
      <Speed value = 1000 unit=”kbps” />    <!-- 1 Mbps, unit Kbps -->
   </DataBus>
   <DataBus>
      <ConnectorType> RJ45 </ConnectorType>       <!-- Jack for EtherNet -->
      <TypePhyMac> EtherCAT </Type>
      <TypeNetTrans> IPTCP </TypeNetTrans>
      <TypeApp> Modbus </TypeApp>
      <Speed value = 10000 unit=”kbps” />    <!-- 1 Mbps, unit Kbps -->
      < DatabusService type = “IDL”>
         Path of IDL file.
      </ DatabusService >
      </DataBus>
  </DataBuses>
   <DBType> SQL DBMS </DBType> <!-- DBMS type    -->
  <IP> IP23CH    </IP>      <!--    Ingress Protection IEC 60529-->
  <Middleware> OPRoS    </Middleware>
  <CommInterface> SERVER-CLIENT </CommInterface>
</Infra>
```

보기 2 IDL형식으로 정의한 Infra의 DataBus클래스의 함수정의는 다음과 같다.

```
interface DatabusService: CIMService
{
        int16 initialize ();// 데이터 획득을 위한 하드웨어 장치를 초기화하는 함수
        int16 open ();// 데이터획득을 위한 하드웨어 장치와 연결하는 함수
        int16 close ();// 데이터 획득을 위한 하드웨어 장치와 연결을 해제하는 함수
        int16 read();// 연결된 하드웨어로부터 데이터를 읽어오는 함수
        int16 write();// 연결된 하드웨어에 데이터값을 인가하는 함수
        int16 control ();// 연결된 하드웨어를 구동시키는 함수
}
```

모듈의 통신방법은 commInterface에 명시를 하는데, 상위 표준인 KOROS 1149-1:2020의 7장 소프트웨어 모듈의 요구사항을 반영한다, 모듈의 통신방법은 4가지타입으로 명시한다: publish/subscribe 타입은 PUB-SUB, client/server 타입은 SERVER-CLIENT, 공유메모리 타입은 SHM, 원격프로시저호출은 RPC 로 명시한다.
모듈이 지원하는 미들웨어를 명시하는데 추가 가능한 미들웨어는 다음과 같다: OPRoS, ROS, OpenRTM, OROCOS 등이 있다.

Class Infra

// 데이터버스
typedef DataBus[] DataBusList;
+ noBuses : int8 // 모듈안에서 사용되는 데이터버스갯수. 기본값 1.
+ dataBus: DataBusList
// 데이터베이스 시스템의 유형
+ dbType : String // 모듈안에서 사용되는 DBMS 유형 예) oracle, sql, …
+ middleware : Enumeration //추가된 미들웨어 열거형 {OPRoS, ROS, OpenRTM, OROCOS}
+ commInterface : Enumeration //추가된 통신방법 열거형 {PUB-SUB, SERVER-CLIENT, SHM, RPC}

그림 4.11 — CIM의 Infrastructures클래스 확장한 Infrastructure 클래스 다이어그램

Class DataBus

typedef String[] Stringlist
+ String connectionType; // 연결타입:USB-A, USB-C, USD-C, RG45, DE9
+ String typePhyMac; // 물리적/ MAC 프로토콜 타입: USB, CAN2.0, EtherCAT, Ethernet, RS485
+ String typeNetTrans; // 네트워크 프로토콜과 전송 프로토콜, TCP/IP, CANopen
+ Stringlist typeApp; // 세션/표현/응용 프로토콜: 프로토콜의 열거형
+ float32 speed; // 전송속도 (단위: Kbps)

+ initialize() : int16 // 확장함. 데이터 획득을 위한 하드웨어 장치의 초기화 + open() : int16 // 확장함 데이터 획득을 위한 하드웨어 장치 연결 + read() : int16 // 확장함 하드웨어 장치로부터 데이터 획득 + write() : int16 // 확장함 하드웨어 장치로 데이터 전달 + close() : int16 // 확장함 데이터 획득을 위한 하드웨어 장치와 연결 해제 + control() : int16 // 확장함 데이터 획득을 위해 연결된 하드웨어 장치의 구동

그림 4.12 — SM의 Infra클래스의 속성인 DataBus 클래스 다이어그램

4.2.9 소프트웨어 모듈의 안전과 보안을 위한 클래스

모듈이 제공하는 안전레벨과 보안레벨를 나타내는 SafeSecurity클래스는 상위 표준인 KOROS 1148:2021의 SafeSecure클래스를 상속받아 확장하여 그림 4.13과 같아야한다. 보안의 측정은 보안의 타입과 보안의 레벨(0~5)을 사용하여 명시해야한다. 모듈은 0이상의 값을 포함할 수 있다. 측정을 나타내는 CyberSecurity클래스를 사용하여 SafeSecurity클래스는 그림 4.13과 같이 나타내야한다.

비고 서비스클래스가 제공하는 안전관련 함수와 보안관련 함수는 안전과 보안 관리자가 사용할 수 있다.

보기 1 XML로 나타낸 SafeSecure클래스 정보의 예는 다음과 같으며, 부속서 A.9에 명시방법을 제공

한다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SafeSecure>
  <SafetyLevel>
    <Overall> d </Overall>          <!-- choose one PL among a, b, c, d, e, and none.    -->
    <SafetyFunction type = “ESTOP “ value=’d’ /> <!-- individual safety measures -->
    <SafetyFunction type = “SRSC “    value = ‘e’ />
  </SafetyLevel>
  <SecurityLevel>
    <CyberSecurityLevel>      <!-- list security functions provided by module -->
      <Overall>      2    </Overall>    <!-- overall security level of a module -->
      <CyberSecurity type = “HU_IA”         value=2 /> <!-- individual security fct -->
      < CyberSecurity type = “ACNT_MGT”    value = 2    />
    </CyberSecurity>
  </SecurityLevel>
  < SafeSecureService type = “IDL”>
    Path of IDL file.
  </ SafeSecureService >
</SafeSecure>
```

보기 2 IDL형식으로 정의한 SafeSecure 클래스의 함수정의는 다음과 같다.

```
interface SafeSecureService: CIMService
{
        Enumeration getSafetyLevel ();// safetyLevel 값을 반환하는 함수
        Boolean setSafetyLevel (Enumeration insl);// safetyLevel에 값을 할당하는 함수
         // 요청된 보안타입의 값을 반환하는 함수
        Enumeration getCybSecurityLevel (securityType str);
         // 요청된 보안타입의 값을 설정하는 함수
        Boolean setCybSecurityLevel (CyberSecurity cybSec);
}
```

Class SafeSecure

typedef SafeytLevel enumeration of {n, a, b, c, d, e, 0, 1, 2, 3, 4 } // PL or SIL
typedef SecurityLevel enumeration of {0,1,2,3,4 }
+ safetyLevel : SafetyLevel // 모듈의 전체적 PL값 a, b, c, d ,e, or n ; 안전레벨없음 ‘n’
// 또는 SIL 값 0,1,2,3, or 4; 안전레벨없음 0
+ cybSecurityLevel: Enumeration // 모듈의 전체적 사이버보안값 0,1,2,3,4
typedef SafetyFunction[] SafetyFunctionList;
+ inSafetyLevel: SafetyFunctionList; // 개별 안전함수의 PL값
typedef CyberSecurity[] CyberSecurityList;
+ inCybSecurityLevel: CyberSecurityList; // 개별보안함수

+ getSafetyLevel() : Enumeration
+ setSafetyLevel(inSL: Enumeration) :Boolean
+ getCybSecurityLevel(securityType:String) : Enumeration
// securityType이 정의되는 경우 값을 반환
// SIL도 같음, 정의되지 않으면 0값 반환
+ setCybSecurityLevel(cybSec: CyberSecurity) :Boolean

그림 4.13 — CIM의 SafeSecure클래스를 확장한 SafeSecure 클래스 다이어그램

4.2.10 소프트웨어 모듈의 모델링을 위한 클래스

소프트웨어 모듈의 시뮬레이션을 위한 정보는 KOROS 1148:2021에 준한다. 제공할 모델링을 위한 정보가 없다면 해당 사항에 “NA“ 라고 명시해야한다.

부속서 A
(규정)

소프트웨어 정보모델 표기법

A.1 개요
부속서는 4장의 클래스의 인스턴스를 생성하는 정보를 표현한다. 정보들은 XML or JSON 으로 표시할 수 있다.본 문서에서는 정보는 XML 표시한다.

A.2 소프트웨어 모듈의 일반정보
소프트웨어 모듈의 일반정보를 위한 XML 요소는 KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부의 부속서C.2를 참조한다

A.3 소프트웨어 모듈의 아이디 정보
소프트웨어 모듈의 아이디 정보를 위한 XML 요소는 KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부의 부속서C.3를 참조한다. 모듈의 아이디 정보를 위한 XML 요소는 'SWAspects'요소 만 가지며, 'HWAspects'요소를 가지지 않는다. 추가되는 XML의 요소는 표 A.1과 같다.

표 A.1 모듈 아이디에 추가되는 XML 요소

요소이름	설명
IDnTypeService	모듈이 구동중에 외부의 모듈에 접근 제한자가 private인 변수인 인스턴스 아이디의 값을 제공하는 함수의 명세방법이 IDL인지 XML인지를 명시한다.

A.4 소프트웨어 모듈의 실행 파일 정보
소프트웨어 모듈의 실행파일을 위한 XML 표 A.1과 같다.

표 A.2 실행 파일의 XML 요소

요소이름	설명
MainFileURL	소프트웨어 모듈을 동작시키는 파일의 이름으로 경로를 포함할 수 있다.
LibraryURL	모듈이 동작하는 필요한 라이브러리들이 있는 경로의 이름

A.5 소프트웨어 모듈의 속성 입력 출력 정보
속성, 입력 및 출력 정보를위한 XML 요소는 KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부의 부속서C.4를 참조한다. 소프트웨어 모듈을 위해 추가되는 XML요소는 표 A.2과 같다.

표 A.3 속성, 입력, 출력에 추가되는 XML 요소

요소이름	설명
Language	소프트웨어 모듈이 작성된 프로그래밍 언어를 명시하여 동작에 필요한 환경을 알수있도록 한다. (예 C/C++, Java, Python)
OSType	모듈이 동작하는 운영체제의 이름을 명시한다. (예 x86, x86_64)

CompilerType	모듈이 빌드된 컴파일러의 정보를 제공하여 동작에 필요한 환경을 알 수 있도록 한다.
CPUType	모듈이 동작하는 CPU를 명시한다. (예 8, 16, 32, 64)
IOVariableService	모듈이 구동중에 외부의 모듈에 접근 제한자가 private인 변수인 ioVariable의 값을 제공하는 함수의 명세방법이 IDL인지 XML인지를 명시한다.

A.6 소프트웨어 모듈의 상태 정보

소프트웨어 모듈의 상태 정보를위한 XML 요소는 표 A.4와 같다.

표 A.4 모듈 상태에 추가되는 XML 요소

요소이름	설명
ExecutionStatus	소프트웨어 모듈의 생명주기 상태의 값을 가진다.
ErrorType	소프트웨어 모듈의 에러값을 가진다.
StatusService	모듈이 구동중에 외부의 모듈에 접근 제한자가 private인 변수인 ExecutionStatus나 ErrorType의 값을 제공하는 함수의 명세방법이 IDL인지 XML인지를 명시한다.

A.7 소프트웨어 모듈의 서비스 정보

소프트웨어 모듈의 서비스 정보를위한 XML 요소는 KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부의 부속서C.5를 참조한다

A.8 소프트웨어 모듈의 인프라 정보

소프트웨어 모듈의 인프라 정보를위한 XML 요소는 KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부의 부속서C.6를 참조한다.

표 A.5 인프라에 추가되는 XML 요소

요소이름	설명
Middleware	소프트웨어 모듈이 연동되는 서비스 로봇 미들웨어의 이름을 명시한다. (예 OPRoS, ROS, OpenRTM, OROCOS)
CommInterface	모듈이 하드웨어 장치로부터 값을 획득하는 경우 사용되는 통신방법을 명시한다. (예 PUB-SUB, SERVER-CLIENT, SHM, RPC)
DatabusService	모듈이 하드웨어 장치로부터 값을 가지고 오기 위해 사용되는 함수의 명세방법이 IDL인지 XML인지를 명시한다.

A.9 소프트웨어 모듈의 안전과 보안의 정보

소프트웨어 모듈의 안전과 보안 정보를위한 XML 요소는 KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부의 부속서C.7를 참조한다

표 A.6 모듈 안전과 보안에 추가되는 XML 요소

요소이름	설명

SafeSecureService	모듈이 구동중에 외부의 모듈에 접근 제한자가 private인 변수인 안전등급 및 보안등급의 값을 제공하는 함수의 명세방법이 IDL인지 XML인지를 명시한다.

A.10 소프트웨어 모듈의 모델링 정보

소프트웨어 모듈의 시뮬레이션 정보를 위한 XML 요소는 KOROS 1148:2021 서비스 로봇을 위한 모듈 – 제2-1부의 부속서C.8를 참조한다

지능형로봇표준포럼

표준서의 서식 및 작성방법

Rules for the drafting and presentation of Korea Robot Standards

KOROS XXXX : YYYY

제 정 자 : 지능형로봇표준포럼 의장

제정 : YYYY년 MM월 DD일

지능형로봇표준포럼 사무국
서울시 용산구 한강대로 31 금영빌딩 8 층한국로봇산업협회
전화 : (02) 780-3060