

## 프로덕트라인 기반의 USN 응용개발을 위한 UML 프로파일

이우진<sup>1</sup>, 최일우<sup>2\*</sup>

<sup>1</sup>세종대학교 정보통신공학과, <sup>2</sup>강남대학교 교양학부

### A UML Profile for USN Application Development based on Software Product Line Approach

Woo-Jin Lee<sup>1</sup> and Il-Woo Choi<sup>2\*</sup>

<sup>1</sup>Dept. of Information and Communication Engineering, Sejong University

<sup>2</sup>Division of General Studies, Kangnam University

**요 약** USN(Ubiquitous Sensor Network) 응용 S/W는 다양한 타겟 운영체제의 핵심모듈들을 기반으로, 다양한 종류의 센서 노드들을 유기적으로 제어하는 복잡한 특징을 가진다. 현재 USN 응용 분야에서도 효율적으로 S/W를 개발하기 위한 다양한 연구가 진행되고 있다. S/W의 개발생산성을 높이기 위해서는 프로덕트라인 기반 개발과 같이 도메인에 따라 어플리케이션들의 핵심 공통기능을 명세하고, 어플리케이션의 워크플로우에 따라 핵심 공통기능에 가변적인 기능만을 취사 선택하여 개발하는 방법이 효율적이다. 이러한 방법을 USN 도메인에 적용하기 위해서는 USN 응용 S/W의 특성과 프로덕트라인 기반 개발의 특성을 일관성 있는 하나의 뷰로 명세 가능하여야 한다. 그러나 일반적인 UML 표기법만으로는 이러한 특성을 효과적으로 명세하기 어렵다. 본 논문에서는 이러한 문제점을 해결하기 위하여 기존 UML의 확장 메커니즘인 프로파일을 이용하여 USN 어플리케이션의 특성과 프로덕트라인 기반 개발의 특성을 효과적으로 명세할 수 있는 기법을 제시한다.

제시하는 프로덕트라인 기반 USN 응용 개발을 위한 UML 프로파일은 어플리케이션 개발자에게 USN이나 프로덕트라인 기반 개발에 특화된 새로운 설계기법이나 도구 등의 추가적인 자원을 요구하지 않고, 기존의 UML과 지원도구를 활용하여 효과적으로 프로덕트라인 기반의 USN 응용 개발을 가능하게 한다.

**Abstract** USN(Ubiquitous Sensor Network) application S/W has a complex characteristic that it controls various kind of sensor nodes organically based on the core modules of various target operating systems. Currently, various researches for efficient S/W development are being performed in USN application field. In order to increase S/W productivity, the method such as product line development approach to specify core common functions of applications according to the domain and develop applications with the core common functions and variant functions selected according to the workflow of applications is efficient. To apply this approach to the USN domain, characteristics of USN application S/W and product line development should be specified with one consistent view. However, it is difficult to efficiently specify the characteristics using general UML notations. In order to solve the problem, this paper proposes a method to effectively specify the characteristics of USN applications and product line development using UML profile which is the extension mechanism of UML. The proposed UML profile for produce line based USN application development enables developers to effectively develop product line based USN applications using existing UML and UML support tools without new design methods or tools which are specified for USN or product line development.

**Key Words** : Sensor Network, UML profile, Product Line

---

이 논문은 강남대학교 교내연구비 지원을 받아 연구 된 것임.

\*Corresponding Author : Il-Woo Choi

Tel: +82-31-280-3662 email: iwchoi@kangnam.ac.kr

접수일 12년 04월 30일

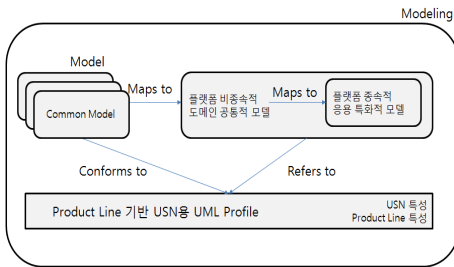
수정일 12년 06월 08일

게재확정일 12년 09월 06일

## 1. 서론

USN(Ubiquitous Sensor Network)[1]은 현재 다양한 응용 분야에서 적극 활용 중이다. 센서네트워크는 다양한 역할을 수행하는 센서 노드들로 구성된다. 센서 노드들은 자원이 한정되어 있으므로 효율적으로 USN 응용 S/W를 개발하기 위해서는 불필요한 모듈을 노드에 탑재하지 않고 필수모듈만을 선택적으로 탑재해야 한다. 그러므로 USN 응용 S/W는 모듈 중 노드들에 필요한 핵심 공통부분과 특성에 따라 다양한 역할 수행을 위한 가변부분을 구분하여 접근하는 것이 효율적이다.

프로덕트라인(Product Line)[2]기반 개발은 이러한 USN 응용 S/W의 특성에 적합한 개발기법 중 하나이다. 핵심모듈의 공통적인 부분을 핵심자산(Core Asset)으로 구축하여 재사용하고, 다양한 디바이스와 기능에 따라 가변적인 부분을 선택적으로 재사용하여 복잡한 USN 응용 S/W 개발을 효율적으로 지원할 수 있기 때문이다. USN 응용 S/W 개발에 효율적으로 프로덕트라인 기반 개발기법을 적용하기 위해서는 복잡한 USN 응용의 특성과 함께 피쳐, 공통성, 가변성 등의 프로덕트라인 기반 개발의 특성을 일관성 있는 하나의 뷰로 명세 가능해야 한다. 그러나 현재의 업계명세표준으로 자리 잡고 있는 UML의 일반적인 표기법만을 가지고는 이러한 표현이 불가능한 실정이다.



[그림 1] 프로파일의 역할  
[Fig. 1] The Role of UML Profile

본 논문에서는 USN 응용 도메인에 효율적으로 프로덕트라인 기반 개발기법을 적용, 명세하기 위한 UML 프로파일을 제시한다. UML 프로파일[3, 4]은 UML의 확장 메커니즘으로 스테레오타입(Stereo type), 꼬리표 값(Tagged value), 제약사항(Constraint) 등의 확장요소로 구성되는데, 이를 이용하여 USN의 특성과 프로덕트라인 기반 개발의 특성을 확장요소로 정의하고, 일관성 있는 하나의 뷰로 명세 가능하게 한다. 확장된 요소들은 명세를 위하여 해당하는 구문(Syntactic), 의미(Semantic), 규

약(Contract) 및 표기법을 가진다. 제시하는 프로덕트라인 기반 USN용 프로파일의 경우 그림 1과 같이 일반적인 UML 모델이 프로파일의 명세 장치를 준수(Conforms to)하여야 한다. 핵심자산 구축을 위한 플랫폼 비종속적, 도메인 공통적 모델 및 USN 특성을 반영한 플랫폼 종속적, 응용 특화적 모델로 변환될 때의 변환규칙 및 변환도구는 이 프로파일을 참조(Refers to)하게 된다.

제시된 프로파일은 USN 응용 도메인에 피쳐, 공통성, 가변성 등의 프로덕트라인 개발 특성을 적용 가능하게 하고 USN에 국한된 다양한 특성에 대한 결정사항을 반영하게 한다. 이는 UML을 통하여 USN의 특성 및 프로덕트라인의 특성을 일관성 있는 하나의 뷰로 명세 가능하게 한다.

## 2. 관련연구

### 2.1 USN 개발 특성

현재 USN 응용 S/W를 효율적으로 개발하기 위하여 다양한 기법 및 도구들이 연구, 활용되고 있다. 그 중에서도 특히 TinyGALS[5], AtaG[6], ABD[7] 등은 본 논문에서 제시하고자 하는 방향과 같이 모델로부터 USN 응용 S/W를 자동으로 생성하기 위한 방법을 제공한다. 이러한 기종방법들은 모델로부터 쉽고, 빠르게 USN 응용 S/W를 생성할 수 있게 해 준다는 장점이 있으나 범용적이지 않은, USN 응용 도메인에 특화된 방법이라는 한계를 지닌다. 이렇게 특화된 기법들을 사용하는 이유는 USN 어플리케이션이 다양한 타겟 운영체제를 기반으로, 다양한 역할을 수행하는 노드가 유기적으로 연관되므로, 플랫폼에 종속적으로 노드들의 역할 및 관계 등을 명세해야 하는 특성 때문이다. 이러한 USN 어플리케이션의 특성이 분석, 설계단계에서 명세에 반영되어야 하므로 보편적 표준 명세언어인 UML의 활용에는 어려움이 있다.

### 2.2 프로덕트라인과 USN개발

프로덕트라인[2] 기반 개발기법은 한 프로덕트라인에 속하는 여러 어플리케이션들이 서로 공유할 수 있는 핵심자산을 재사용하는 대단위 S/W재사용방법으로 FOPL (Feature Oriented Product Line Engineering)[8], PuLSE (Product Line Software Engineering)[9]등 다양한 연구와 적용이 이루어져 왔다. 프로덕트라인 기반 개발기법은 유사 도메인의 기능적 공통성과 가변성에 기반하여 공통기능을 포함하는 핵심자산 개발과 이를 이용한 프로덕트 개발의 측면을 포함하고 있다. 핵심자산은 프로덕트라인

의 여러 멤버에서 재사용될 수 있기 때문에, 공통성과 가변성의 명확한 정의는 높은 재사용성을 제공, 생산성을 향상시켜 고품질의 어플리케이션을 빠른 시간 내에 개발하는데 필수 요소이다. USN 응용S/W의 효율적인 개발을 위해서는 핵심모듈의 공통기능과 수많은 센서 노드들의 다양한 역할 수행을 위한 가변기능을 구분하여 생산성을 높일 필요가 있다. 그러나 기존의 USN 분야에 특화된 명세방법으로는 피처, 공통성, 가변성 등의 특성명세가 불가능하다. 또한 프로덕트라인 기법의 명세방법을 차용하여 핵심자산을 구축하더라도 프로덕트라인 기법의 특성과 USN응용의 특성을 반영하여 일관성 있는 하나의 뷰로 명세할 수 있는 방법이 마련되어 있지 않아 하나의 시스템을 서로 다른 여러 관점의 뷰와 상이한 노테이션을 통하여 명세하여야 하는 복잡한문제가 발생한다[12].

### 3. 프로파일정의

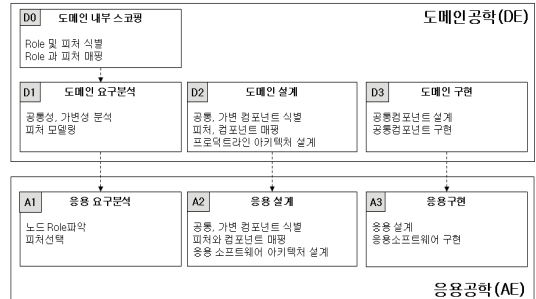
본 장에서는 프로덕트라인 기반 USN용 UML 프로파일의 정의를 위하여, 프로덕트라인 기반 USN 응용 개발 프로세스에 따라 프로덕트라인의 특성이 반영되는 도메인공학(Domain Engineering)과 USN의 특성이 반영되는 응용공학(Application Engineering) 부분으로 나누어 프로파일을 정의, 설명한다.

#### 3.1 프로덕트라인 기반 USN 개발 프로세스

본 논문의 프로덕트라인 기반 USN 응용 S/W 개발 절차는 기존의 프로덕트라인 개발 프로세스[10]를 도메인의 특성에 알맞게 커스터마이징하여 그림 2와 같이 구성하였다.

프로세스는 크게 도메인공학과 응용공학으로 구분된다. 각 단계는 요구분석, 설계, 구현의 공통된 3단계로 구성되며, 도메인공학에서는 요구분석 이전에 도메인 내부 스코핑 단계를 수행한다. 이 단계에서는 기능-역할 매핑을 통하여 식별한 USN 응용 S/W의 피처들과 응용 S/W 특성과의 관계를 명세 할 수 있으며, 요구분석 단계에서는 USN 응용 S/W가 가지는 공통성과 가변성을 찾아내고, 가변성 피처를 명세한다. 설계 단계에서는 USN 응용 S/W의 각 피처들을 구현 컴포넌트로 매핑하며, 이를 바탕으로 응용 S/W에 대한 프로덕트라인 아키텍처를 설계한다. 그리고, 구현단계에서는 설계 단계에서 작성한 아키텍처에 따라 기능-역할, 속성-역할 매핑[11]을 활용, 운영체제, 구현 언어, 응용특성 등을 고려하여 USN 응용 S/W를 명세, 구현한다. 결과적으로 도메인공학에서는 프

로덕트라인의 특성들이, 응용공학에서는 USN의 특성들이 명세에 집중적으로 반영된다.



[그림 2] 프로덕트라인기반 개발 프로세스  
[Fig. 2] The Development Process based on Product Line Approach

#### 3.2 프로덕트라인 특성 정의

프로덕트라인 특성은 제시한 프로세스 중 도메인공학 단계(DE)에서 주로 고려된다. 본 절에서는 UML을 이용하여 효과적으로 프로덕트라인을 설계할 수 있도록 적용할 프로덕트라인의 특성[8]을 UML 프로파일의 확장요소로 정의한다.

도메인 내부 스코핑 단계(D0)에서는 USN의 다양한 노드들을 위한 응용 S/W가 갖게 될 피처들을 파악하고, 응용 S/W의 특성에 따라 피처들을 매핑한다. 하나의 센서네트워크는 수많은 종류의 노드들로 구성되고, 응용 도메인에 따라 노드들의 종류가 달라질 수 있으므로, 도메인 내부 스코핑을 통하여 프로덕트라인에 포함될 응용 S/W의 종류와 피처들을 파악하는 것이 필요하다.

[표 1] 기능-역할 매핑테이블  
[Table 1] Function-Role Mapping Table

기능 \ 역할		역할			
		Sensor	Router	Sink	Actuator
기능 #1	Type 1-1	✓			
	Type 1-n	✓			
기능 #2	Type 2-1		✓	✓	
	Type 2-2		✓	✓	
	Type 2-n	✓			✓

피처그룹과 피처의 도출은 각 노드의 기능은 타겟 운영체제의 모듈을 사용하여 구성되므로 타겟 운영체제에 따라 서로 다른 기능이 제공된다. 따라서 기능은 타겟 운영체제에 따라 표 1과 같은 기능-역할 매핑을 통하여 도

출할 수 있다. 기능은 기능유형을 통하여 더 세분화할 수 있는데 기능은 피처그룹으로 기능유형은 피처로 매핑된다.

도메인 요구분석 단계(D1)에서는 도메인 내부 스크핑 단계의 결과를 분석하여 응용 S/W 피처들의 공통성 및 가변성을 결정하고, 가변성 피처모델을 명세한다.

*Rule 1: 추출된 피처그룹 혹은 피처를 클래스로 매핑하고 공통성, 가변성 등의 특성에 따라 스테레오타입으로 정의한다.*

피처, 공통성, 가변성 등의 프로덕트라인의 특성을 명세하기 위하여 적용할 UML 확장요소는 Rule 1 같은 기준으로 정의한다.

표 2는 가변성 피처모델에서 지원 가능한 다양한 가변성 표기법을 나타낸다. 가변성 피처모델의 표기법은 UML의 표기법을 바탕으로 사용자들이 쉽게 USN 응용 S/W에 대한 피처모델을 작성할 수 있도록 스테레오타입을 확장, 정의하였다.

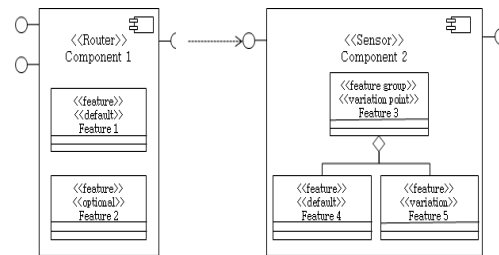
하나의 피처 또는 피처그룹은 하나 혹은 그 이상의 클래스로 매핑되며, 각 피처 혹은 피처그룹들은 표 2의 확장요소들을 통하여 특성을 명세한다.

«zero-or-one feature group»은 필요에 따라 그룹에 속해 있는 피처가 선택되지 않거나 하나만 선택되어야 하는 경우를 나타낸다. «exclusive-or feature group»은 그룹에 속해 있는 피처 중에서 오직 하나만 선택되어야 하는 경우를 나타낸다. «inclusive-or feature group» 필요에 따라 그룹에 속해 있는 피처가 선택되지 않거나 하나이상 선택되는 경우를 나타낸다. «variation point»는 가변점을

나타낸다. USN 응용 S/W의 특성에 따라 가변적인 피처들의 그룹을 표현할 때 사용한다.

«optional» 피처타입은 필요에 따라 포함되는 피처를 나타낸다. «alternative»는 선택적인 피처를 나타낸다. «default»는 기본으로 포함되는 피처를 나타낸다. «requires»는 피처 사이의 관계를 나타내는 것으로, 하나의 피처가 선택되면 같이 선택되어야 하는 피처 간의 관계를 나타낼 때 사용한다. «variant»는 가변적인 피처를 나타낸다.

도메인 설계 단계(D2)에서는 공통의 피처에 대한 컴포넌트를 식별하고, USN 응용 S/W에 대한 프로덕트라인 아키텍처를 명세한다. USN 응용 S/W를 구현하기 위해서는 응용 S/W를 구성하는 컴포넌트를 식별하고, 응용 S/W의 피처를 반영하여 각 컴포넌트를 명세해야 한다. 이를 위해서 도메인 설계 단계에서는 도메인 요구분석 단계(D1)에서 파악한 공통된 피처를 응용 S/W의 컴포넌트로 매핑한다.



[그림 3] 피처와 컴포넌트 매핑 사례  
[Fig. 3] An Example of Feature-Component Mapping

그림 3은 피처-컴포넌트 매핑 사례를 보여준다. 일반

[표 2] 피처, 공통성, 가변성 표현을 위한 확장요소  
[Table 2] Extensions for Features, Commonality and Variations

표기법	확장유형요소	설명
<pre> classDiagram     class FeatureGroup {         &lt;&lt;Feature&gt;&gt;         Feature     }                     </pre>	Class	피처 노테이션
<<feature group name>>	Stereotype	피처그룹 식별자
<<feature name>>		피처 식별자
<<feature group type>>		피처그룹의 타입 <<zero-or-one feature group>>, <<exclusive-or-one feature group>>, <<inclusive-or-one feature group>>, <<variation point>>
<<feature type>>		피처타입 <<optional>>, <<alternative>>, <<default>>, <<requires>>, <<variant>>

적으로 하나의 컴포넌트는 하나 이상의 피처그룹과 피처를 반영하여 명세된다. 공통된 피처를 컴포넌트로 매핑한 후에는 컴포넌트들을 구성하여 응용S/W의 프로덕트 라인 아키텍처를 명세한다. 이 단계에서는 가변성 피처모델에서 제시된 가변성을 아키텍처 관점으로 일관성 있게 반영하기 위하여 가변성 표기를 추가하여 명세한다. 도메인공학 단계에서는 USN 응용 S/W의 공통부분에 대해서만 컴포넌트로 명세되고, 가변적인 부분에 대해서는 «variation point»을 사용하여 남겨둔다. «variation point»에 해당하는 부분은 응용공학 단계에서 실제 개발할 S/W의 특성에 맞도록 선택한 피처를 반영하는 컴포넌트로 대체가 된다.

### 3.3 USN 특성 정의

USN의 특성은 프로세스 전 단계에 걸쳐서 명세에 반영되지만 특히 응용공학(AE)에 집중된다. UML을 이용하여 효과적으로 USN 어플리케이션을 명세할 수 있도록 USN의 특성을 UML 확장요소로 정의한다.

#### 3.3.1 USN 어플리케이션 구성요소

USN은 다양한 역할을 수행하는 노드가 서로 유기적으로 연결되어 있는 것이다. 따라서 USN 어플리케이션은 노드를 구성하는 S/W와 노드간의 관계에 따른 S/W의 유기적 연결로 구성된다. 이에 따라 USN 어플리케이션 모델은 노드의 역할에 따른 노드 S/W와 노드간의 데이터 전송 관계를 포함한다. USN 어플리케이션은 노드와 그들 간의 관계로 정의되며, 노드는 USN을 구성하는 각 노드의 역할에 따라서 센서, 라우터, 싱크, 액츄에이터 등의 기능을 수행하기 위해 필요한 속성과 기능을 가진다. 노드간의 관계는 노드 S/W 간의 데이터 전송 관계를 표현한다[11].

응용설계 단계(A2)에서 UML을 이용하여 명세하는 경우, 컴포넌트 다이어그램, 클래스 다이어그램을 이용하여 노드는 컴포넌트로, 피처는 클래스로, 기능과 속성은 클래스 및 클래스의 오퍼레이션과 애트리뷰트로, 노드간의 관계는 연관관계로 각각 매핑되어 USN 어플리케이션의 특성을 표현한다. USN의 특성을 프로파일로 정의하기 위하여 기능-역할, 속성-역할 매핑을 수행하고, 이를 통하여 컴포넌트, 클래스, 오퍼레이션, 애트리뷰트에 적용할 확장요소를 정의한다.

#### 3.3.2 컴포넌트관련 확장요소정의

USN 어플리케이션 설계 시, 노드 S/W는 UML의 컴포넌트로 매핑한다. 이 때, 각 노드 S/W는 노드의 역할에


따라 포함하는 속성 및 기능이 달라지게 된다. 따라서 노드 S/W의 역할을 구분하기 위한 UML 확장요소가 요구된다. 노드 S/W의 역할을 구분하기 위해 컴포넌트에 적용할 UML 확장요소는 다음의 Rule 2와 같은 기준으로 정의한다.

**Rule 2:** 노드의 역할을 컴포넌트에 적용할 스테레오타입으로 정의한다.

표 3은 컴포넌트에 적용하기 위해 새로 정의한 스테레오타입을 나타낸다. 센서네트워크를 구성하는 노드의 역할은 타겟 운영체제와 관련 없이 유사하므로 «Sensor», «Router», «Sink», «Actuator» 그리고 복합적인 기능을 수행하는 «Composite»로 정의할 수 있다.

[표 3] 컴포넌트에 적용할 확장요소

[Table 3] Extensions for Component

표기법	확장유형 요소	설명
	Component	노드 노테이션
<<Type Name>>	Stereotype	노드 타입 <<Sensor>>, <<Router>>, <Sink>>, <<Actuator>>, <<Composite>>

#### 3.3.3 클래스관련 확장요소정의

USN 어플리케이션 설계 시, 노드 S/W의 기능과 기능요소는 UML의 클래스와 오퍼레이션으로 매핑된다. 이 기능과 기능요소들은 대부분 도메인공학(DE) 과정에서 피처그룹과 피처로 추출되며 추출된 피처그룹과 피처들 중 선택된 피처그룹 및 피처의 구현에 클래스관련 확장요소가 적용된다.

노드 S/W의 기능은 타겟 운영체제의 모듈을 사용하여 구성되므로 타겟 운영체제에 따라 서로 다른 기능이 제공된다. 따라서 기능은 타겟 운영체제에 따라 표 1과 같은 기능-역할 매핑을 통하여 도출할 수 있다. 기능-역할 매핑의 각 기능과 기능을 세분화한 기능유형은 USN 어플리케이션 설계를 위한 UML의 클래스와 오퍼레이션에 적용할 확장요소로 정의되는데, 다음 Rule 3, 4와 같은 기준으로 정의한다.

**Rule 3:** 기능-역할 매핑의 각 기능은 선택된 피쳐, 즉 클래스에 적용할 스테레오타입으로 정의한다.  
**Rule 4:** 기능-역할 매핑의 각 기능유형은 선택된 피쳐, 즉 클래스의 오퍼레이션에 적용할 스테레오타입으로 정의한다.

[표 4] 클래스에 적용할 확장요소의 예  
 [Table 4] Extensions for Class

표기법	확장유형 요소	설명
<<Scheduler>>	Stereotype	스케줄러 기능
<<UART>>		UART 기능
<<RF>>		RF통신 기능
<<Action>>		장치 구동 기능
<<ADC>>		데이터 센싱 기능
:		:

[표 5] 오퍼레이션에 적용할 확장요소의 예  
 [Table 5] Extensions for Operation

표기법	확장유형 요소	설명
<<FIFO>>	Stereotype	FIFO 스케줄러
<<PreemptionRR>>		PreemptionRR 스케줄러
<<Printf>>		출력UART
<<Scanf>>		입력UART
:		:

표 4는 Rule 3에 의해 클래스에 적용할 확장요소의 일부를 보여준다. 표 5는 Rule 4에 의해 클래스의 오퍼레이션에 적용할 확장요소의 일부를 보여준다. 표 4의 «Scheduler» 스테레오타입으로 정의되는 클래스에서는 표 5의 «FIFO», «PreemptionRR» 스테레오타입으로 정의되는 오퍼레이션을 가질 수 있다.

USN 어플리케이션 설계 시, 노드 S/W의 속성은 UML의 클래스 애트리뷰트로 매핑된다. 노드 S/W의 속성 중에서 네트워크 구성을 위한 속성은 타겟 운영체제에 따라 달라질 수 있다. 또한 노드의 역할에 따라 사용되는 속성이 다르므로 결과적으로 운영체제에 따라, 노드의 역할에 따라 서로 다른 속성이 필요하다. 이는 표 6과 같은 속성-역할 매핑을 통하여 도출할 수 있다.

[표 6] 속성-역할 매핑테이블  
 [Table 6] Property-Role Mapping Table

속성 \ 역할	Sensor	Router	Sink	Actuator
Property #1	✓	✓	✓	✓
Property #2	✓			
Property #3		✓	✓	

속성-역할 매핑을 통하여 도출된 각 속성을 바탕으로 USN 어플리케이션의 애트리뷰트에 적용할 확장 요소는 다음 Rule 5와 같은 기준으로 정의한다.

**Rule 5:** 속성-역할 매핑을 통하여 도출된 노드의 속성들은 노드 컴포넌트에 속한 클래스의 애트리뷰트에 적용할 스테레오타입으로 정의한다.

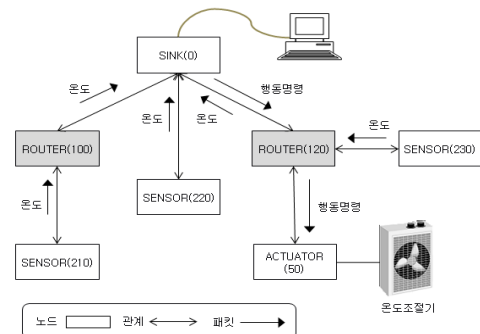
설계 시 운영체제 및 노드의 역할에 따라 적용될 속성은 속성-역할 매핑을 수행하여 표 7과 같이 애트리뷰트의 스테레오타입으로 정의한다.

[표 7] 애트리뷰트에 적용할 확장요소의 예  
 [Table 7] Extensions for Attribute

표기법	확장유형 요소	설명
<<NodeId>>	Stereotype	노드식별자
<<Channel>>		통신채널
:		:

## 4. 적용사례

이 장에서는 본 논문에서 제시하는 프로토타입 기반 USN 응용 개발을 위한 UML 프로파일을 적용한 개발 사례를 제시한다. 제시한 그림 4의 사례는 Nano-Qplus[13] 운영체제를 기반으로 센서네트워크를 통하여 실내온도 제어 기능을 수행한다. 실내온도제어 어플리케이션은 3개의 센서 노드, 2개의 라우터 노드, 1개의 싱크 노드 및 1개의 액추에이터 노드로 구성된다.



[그림 4] 실내온도제어 시스템의 구성도  
 [Fig. 4] The Structure of Indoor Temperature Control System

### 4.1 도메인공학

도메인 내부 스코핑 단계에서는 Nano-Qplus 기반의

센서네트워크에서 필요로 하는 노드들을 위한 응용 S/W가 가져야 할 피처를 찾아내어 공통성 및 가변성을 분석하고, 공통된 부분을 구현하였다.

USN 응용 S/W의 분류는 센서네트워크에 존재하는 노드의 역할에 따라 구분할 수 있다. 센서네트워크에는 크게 센서, 라우터, 싱크, 액추에이터 역할을 하는 노드들이 존재하므로 USN 응용 S/W의 분류 또한 크게 네 가지로 구분할 수 있다. 이와 같이 노드의 역할에 따라 응용 S/W의 분류가 달라지므로, 각 노드의 역할에 따라 응용 S/W에 포함되어야 하는 피처들은 달라진다. 따라서 도메인 내부 스코핑 단계에서는 이러한 것들을 고려하여 USN 응용 S/W가 포함해야 하는 모든 필요한 피처를 추출하였다.

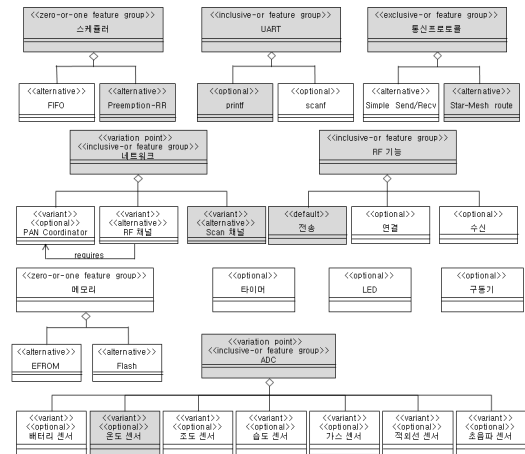
표 8은 Nano-Qplus 기반의 USN 응용 S/W가 가져야 하는 기능과 역할간의 관계를 보여준다. 추출된 피처들은 Nano-Qplus가 제공하는 기능들을 바탕으로 도출한 것이며, 역할은 센서, 라우터, 싱크, 액추에이터 기능을 수행하는 노드를 위한 응용 S/W의 분류를 수행한 것이다.

도메인 요구공학 단계에서는 도메인 내부 스코핑 단계에서 작성한 기능-역할 매핑테이블을 분석하여, 공통성과 가변성을 식별하고, USN 응용 S/W를 개발하기 위한 가변피처 모델을 작성하였다. 표 8을 이용하여 공통성, 가변성을 분석, 그림 5와 같은 가변피처 모델을 작성하였다.

[표 8] 사례의 기능-역할 매핑테이블

[Table 8] Function-Role Mapping Table for Case Study

기능		역할			
기능	기능유형	Sensor	Router	Sink	Actuator
스케줄러	FIFO	✓	✓	✓	✓
	Preemption-RR	✓	✓	✓	✓
UART	Printf	✓	✓	✓	✓
	Scanf	✓	✓	✓	✓
통신 프로토콜	Simple Send/Recv	✓	✓	✓	✓
	Star-Mesh route	✓	✓	✓	✓
네트워크	RF 채널		✓	✓	
	Scan 채널	✓			✓
	PAN coordinator		✓	✓	
RF 기능	연결		✓	✓	
	전송	✓	✓	✓	✓
	수신		✓	✓	✓
메모리	EEPROM	✓	✓	✓	✓
	Flash Memory	✓	✓	✓	✓
타이머		✓	✓	✓	✓
LED		✓	✓	✓	✓
구동기					✓
ADC	온도 센서	✓			
	조도 센서	✓			
	습도 센서	✓			
	가스 센서	✓			



[그림 5] 사례의 가변피처 모델  
[Fig. 5] Variant Feature Model of Case Study

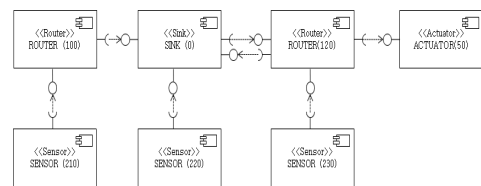
노드의 역할에 따라 가변적인 피처들은 <<variant>> 스테레오 타입을 이용하여 표현하고, 이러한 피처를 포함하고 있는 피처그룹은 <<variation point>> 스테레오 타입을 이용하여 표현함으로써, Nano-Qplus 기반의 센서네트워크 응용 S/W를 위한 피처들의 공통성과 가변성을 명확히 알 수 있는 피처 모델을 작성하였다.

#### 4.2 응용공학

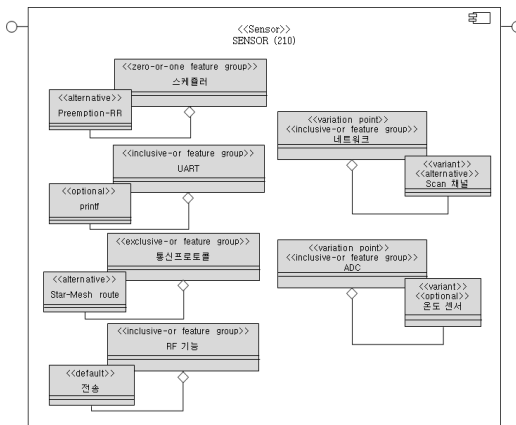
응용공학 단계(AE)에서는 USN 응용 S/W 아키텍처의 가변적인 부분에 대한 컴포넌트들을 상세 설계하여 구현하고, 도메인 공학에서 개발한 공통 컴포넌트들과 연결하여 특정 역할을 수행하는데 필요한 USN 응용 S/W를 완성하였다. 표 2, 3, 4, 5, 6에서 정의한 UML 확장요소를 사용하여 실내용도제어 어플리케이션을 명세하였다.

그림 6, 7은 도메인공학 단계에서 정의한 가변피처 모델을 바탕으로 명세한 컴포넌트 모델이다.

표 9는 표 8에서 도출한 기능-역할 매핑테이블을 바탕으로 Rule 3, 4를 적용하여 클래스와 클래스의 오퍼레이션에 적용할 UML 확장요소를 정의한 것이다.



[그림 6] 사례의 컴포넌트 모델  
[Fig. 6] Component Model of Case Study



[그림 7] 온도센서 컴포넌트를 구성하는 피쳐  
[Fig. 7] Features of Temperature Component

[표 9] 클래스 및 오퍼레이션에 적용할 확장요소  
[Table 9] Extensions for Class and Operation of Case Study

표기법	확장 유형 요소	적용 범주	설명
<<Scheduler>>	Stereotype	Class	스케줄러
<<FIFO>>		Operation	FIFO스케줄러
<<PreemptionRR>>		Operation	PreemptionRR 스케줄러
<<UART>>		Class	UART모듈
<<Printf>>		Operation	출력UART모듈
<<Scanf>>		Operation	입력UART모듈
<<RF>>		Class	RF통신모듈
<<Simple>>		Operation	Simple RF통신모듈
<<IEEE802.15.4MAC>>			MAC RF통신모듈
<<StarMesh>>			StarMesh RF통신모듈
<<Memory>>		Class	메모리제어모듈
<<EEPROM>		Operation	EEPROM 제어모듈
<<FlashMemory>>			플래시메모리제어 모듈
<<Timer>>		Class	타이머 모듈
<<LED>>		Class	LED 제어 모듈
<<Action>>		Class	장치 구동모듈
<<ADC>>		Class	데이터 센싱모듈
<<Temperature>>		Operation	온도 센싱모듈
<<Light>>			조도 센싱모듈
<<Gas>>			가스 센싱모듈
<<Humidity>>	습도 센싱모듈		

에트리뷰트에 적용할 UML의 확장요소를 추출하기 위해 표 10과 같이 속성-역할 매핑 테이블을 작성하고, Rule 5에 의하여 표 11과 같이 에트리뷰트에 적용할 다양한 확장요소를 정의하였다.

[표 10] 속성역할 매핑테이블

[Table 10] Property-Role Mapping Table of Case Study

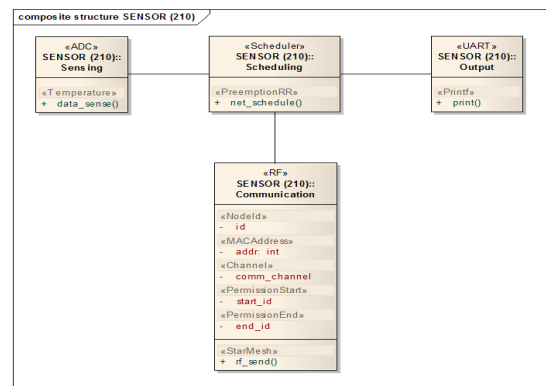
속성	Sensor	Router	Sink	Actuator
노드 식별자	✓	✓	✓	✓
통신 채널	✓	✓	✓	✓
PAN coordinator 노드여부		✓		
근접Actuator 노드의 ID	✓	✓	✓	✓
기본 MAC 주소	✓	✓	✓	✓
확장 MAC 주소	✓	✓	✓	✓
연결허용 시작 노드의 ID	✓	✓	✓	✓
연결허용 끝 노드의 ID	✓	✓	✓	✓
라우팅 1순위 노드의 ID		✓		
라우팅 2순위 노드의 ID		✓		

그림 8은 표 9, 11에서 정의한 확장요소들을 이용하여 센서 노드 SENSOR(210) 컴포넌트에 대한 클래스 다이어그램을 설계한 것이다.

[표 11] 에트리뷰트에 적용할 확장요소

[Table 11] Extensions for Attribute of Case Study

표기법	확장 유형 요소	설명
<<NodeId>>	Stereotype	노드식별자
<<Channel>>		통신채널
<<PAN>>		PANcoordinator여부
<<AdjacentActuator>>		근접액츄에이터 노드ID
<<MACAddress>>		기본MAC주소
<<ExtendMACAddress>>		확장MAC주소
<<PermissionStart>>		연결허용 시작노드ID
<<PermissionEnd>>		연결허용 끝노드ID
<<RoutingFirst>>		라우팅1순위 노드
<<RoutingSecond>>		라우팅2순위 노드



[그림 8] 온도센서 컴포넌트의 클래스 다이어그램

[Fig. 8] Class Diagram for Temperature Sensor Component



### 4.3 평가

[표 12] 기존의 모델기반개발방법과의 비교  
 [Table 12] Comparison with Existing Model-based Development Techniques

	TinyGALS[5]	ATaG[6]	ABD[7]	본 논문
지원플랫폼	단일	멀티	멀티	멀티
설계방법 (다이어그램)	자체 노테이션	자체 노테이션	클래스 일부	클래스 컴포넌트
확장성	낮음	중간	높음	높음
자동화	가능	가능	가능	가능
범용/편이성	낮음	낮음	중간	높음
재사용성	중간	높음	높음	높음

표 12는 기존의 USN 응용 S/W 개발 기법 중에서 관련 연구에서 언급한 모델 기반의 개발 방법과 본 논문의 방법을 비교 평가한 결과이다. 본 논문의 기법은 모델로부터 USN 응용 S/W를 개발하기 위한 방법이므로 기존의 여러 기법 중에서 모델 기반의 개발 방법과 비교하였다.

## 5. 결론

USN 어플리케이션은 다양한 운영체제를 기반으로 수행되고 운영체제의 모듈을 사용하여 노드 소프트웨어가 개발되므로, 어플리케이션 설계시 운영체제, 노드의 역할, 기능, 속성 등 USN 특성에 알맞는 명세를 수행하여야 한다. 그러나 일반적인 UML 모델링을 통하여 설계할 경우, 이러한 결정사항을 반영할 수 없다. 또한 USN 어플리케이션을 개발하기 위한 다양한 방법들은 단일 센서 네트워크 응용 S/W를 빠르게 개발하기 위해 모델이나 추상화된 설계로부터 자동적인 코드를 생성하는데 관심을 두고 있다. 이러한 기존의 방법으로는 핵심공통 모듈의 재사용을 통한 개발 생산성을 극대화하기는 어렵다.

본 논문에서는 프로덕트라인 기법을 USN 응용S/W 개발에 적용, 공통성과 가변성 분석을 통하여 식별된 피쳐 모델을 기반으로 핵심자산을 구축하고, 선택적으로 재사용하여 USN 어플리케이션을 구축하는 사례를 제시하였다. 이를 위하여 프로덕트라인의 특성과 USN 어플리케이션의 특성을 동시에 표현할 수 있는 명세기법이 요구되는데 본 논문에서는 보편적으로 활용되는 UML의 프로파일 메커니즘을 이용, UML의 요소를 프로덕트라인 특성과 USN 어플리케이션 특성을 명세하기에 적합하도록 확장, 정의하였다.

정의한 UML 프로파일을 통하여 개발자는 설계와 도구 활용을 위하여 추가적인 자원을 투자하지 않고, 보편적으로 사용되는 UML과 CASE 도구를 이용하여 USN

어플리케이션을 효율적으로 명세하고 설계할 수 있다. 따라서 기존의 USN에 특화된 기법들에 비하여 범용성 및 사용 편이성이 월등히 높다. 또한 현재 활용중인 다양한 명세 도구들도 표준 UML 플러그인을 사용하므로 USN 용 UML 프로파일을 플러그인으로 배포함으로써 다양한 CASE도구에서 특정 도메인이나 목적에 맞게 개발된 USN 프로파일의 활용이 가능하다.

## References

- [1] Buratti, C., Conti, A., Dardari, D. and Verdone, R., "An Overview on Wireless Sensor Networks Technology and Evolution", *Sensors*, vol.9, no.8, pp.6869-6896, 2009.
- [2] Pohl, K., Bockle, G. and F. van der Linden, "Software Product Line Engineering: Foundations, Principles, and Techniques," Springer, 2005.
- [3] Alhir, S.S., *Guide to Applying the UML*, Springer, 2002.
- [4] "Profile(UML)," [http://en.wikipedia.org/wiki/Profile\\_\(UML\)](http://en.wikipedia.org/wiki/Profile_(UML))
- [5] Cheong, E., Liebman, J., Liu, J. and Zhao, F., "Tinygals: a programming model for event-driven embedded systems", *The Eighteenth Annual ACM Symposium on Applied Computing (SAC 2003)*, 2003.
- [6] Bakshi, A., Prasanna, V. K., Reich, J. and Lerner, D., "The abstract task graph: A methodology for architecture-independent programming of networked sensor systems", *2005 Workshop End-to-end, Sense-and-Respond Systems, Applications and Services (EESR'05)*, 2005.
- [7] Lee, W., Kim, J. and Kang, J., "Automated Construction of Node Software Using Attributes in a Ubiquitous Sensor Network Environment", *Sensors*, vol.10, no.9, pp.8663-8682, 2009.
- [8] Kang, Kyo C., Lee, K., Lee, J. and Kim, S., "Feature-Oriented Product Line Software Engineering: Principles and Guidelines," *Domain Oriented Systems Development: Practices and Perspectives*, Taylor & Francis, page 29-46, 2003.
- [9] Bayer, J., Flege, O., Knauber, P., Laqua, R., Muthig, D., Schmid, K., Widen, T., and Debaud, J., "PuLSE: A Methodology to develop Software Product Lines," *Symposium for software Reusability 99*, 1999.
- [10] Kim, Y., Lee, W. and Choi, I., "Development of

- Ubiquitous Sensor Network Applications based on Software Product Line Approach", KIPS Transactions : Part A, Vol. 14-A, No. 7, pp. 399-408, 2007. 12.
- [11] Lee, W. and Choi, I., "A Method to Define UML Profile for USN Application Design", GESTS International Transaction on Computer Science and Engineering, Vol. 65, No. 1, pp. 95-101, 2011.09.
- [12] Gomaa, H., "Designing Software Product Lines with UML," Addison-Wesley Professional, 2004.
- [13] Lee, K., Shin, Y., Choi, H. and Park, S., "A Design of Sensor Network System based on Scalable & Reconfigurable Nano-OS Platform", In Proceedings of IT-SoC International Conference, 2004.

**이 우 진(Woo-Jin Lee)**

[정회원]



- 2002년 2월 : 숭실대학교대학원 (컴퓨터공학석사)
- 2007년 2월 : 숭실대학교대학원 (컴퓨터공학박사)
- 2009년 3월 ~ 현재 : 세종대학교 정보통신공학과 초빙교수

<관심분야>

센서네트워크, 모바일컴퓨팅, 소프트웨어개발

**최 일 우(Il-Woo Choi)**

[정회원]



- 1997년 2월 : 숭실대학교대학원 (컴퓨터공학석사)
- 2004년 2월 : 숭실대학교대학원 (컴퓨터공학박사)
- 2007년 3월 ~ 현재 : 강남대학교 교양학부 교수

<관심분야>

개발 프로세스, 재사용, SOA, USN, 모바일컴퓨팅