



碩士學位論文

센서 웹 기반의 COST  
설계 및 구현

濟州大學校 大學院

컴퓨터工學科

安 演 峻

2011 年 08 月

# 센서웹 기반의 COST 설계 및 구현

指導教授 金 度 縣

安 演 峻

이 論文을 工學 碩士學位 論文으로 提出함

2011年 08月

安演峻의 工學 碩士學位 論文을 認准함

審査委員長 \_\_\_\_\_ 印

委 員 \_\_\_\_\_ 印

委 員 \_\_\_\_\_ 印

濟州大學校 大學院

2011年 08月

Design and Implementation of COST  
based on the SensorWeb

**Yeon-Jun An**

(Supervised by professor Do-Hyeun Kim)

A thesis submitted in partial fulfillment of the requirement for the  
degree of Master of Computer Engineering

2011. 08.

This thesis has been examined and approved.

Thesis director, \_\_\_\_\_

Thesis director, \_\_\_\_\_

Thesis director, \_\_\_\_\_

August 2011

Department of Computer Engineering  
Graduate School  
Jeju National University

## 감사의 글

2009년 대학원 생활을 시작하여 어느덧 석사 학위과정을 마치고 학위를 받게 되었습니다. 2년 동안 열심히 했던 대학원 생활을 뒤 돌아보니 저에게 도움을 주셨던 모든 분들이 머릿속을 스쳐 갑니다.

먼저 제가 2년간 대학원 생활을 함에 있어 아낌없이 가르침과 지도를 해 주신 지도교수님 이신 김도현 교수님께 감사합니다. 또한 논문 심사에 애써주신 곽호영 교수님과 송왕철 교수님께도 감사합니다. 항상 대학원 생활에 세심한 지도와 가르침을 아끼지 않아주셨던 김장형 교수님, 안기중 교수님, 곽호영 교수님, 이상준 교수님, 변상용 교수님, 송왕철 교수님, 변영철 교수님 정말 감사드립니다. 그리고 대학원에 쉽게 적응하도록 도움을 주신 학과사무실 정은경 선생님, 김남식 선생님 감사드립니다.

대학원에 같이 입학하여 동고동학한 전현탁 선생님, 고영일 선생님, 항상 도움을 주시는 권 훈 선생님, 김남식 선생님, 노영식 선생님, 한대오 선생님, 송재훈 선생님, 김미정 선생님, 손희철 선생님 모두 감사드립니다. 모바일 컴퓨팅 연구실의 이성현, 김세훈, 윤인찬, 윤미자, 정재훈, 진 남씨 함께한 시간이 많지는 않았지만 즐거웠고 모두 감사합니다. 모두들 희망하는 회사에서 근무할 수 있기를 기원하며 정기적으로 교수님과 한 자리에서 볼 수 있었으면 좋겠습니다. 모두들 건강하고 하는 일마다 축복이 있기를 기원합니다.

회사와 학교 두 마리 토끼를 쫓아 정신없었던 시간에 균형을 잡을 수 있도록 많은 도움을 주신 강종원 사장님, 이철식 이사님, 김 혁 팀장님, 손희철 소장님, 강민성 팀장님, 강경석 연구원님 모두 감사드립니다.

끝으로 부족한 저에게 언제나 힘이 되어주며 조언을 해주는 친구 박영석, 고영진, 양의호, 송정근과 항상 제 곁에 있으면서 응원 해 주시고 격려해 주시던 부모님에게 감사함을 전합니다.

# 목 차

그림목차 .....	iii
표목차 .....	v
국문초록 .....	vi
영문초록 .....	vii
약어표 .....	ix
<b>I. 서 론 .....</b>	<b>1</b>
1. 연구배경 .....	1
2. 연구 목적 및 방법 .....	2
3. 논문 구성 .....	3
<b>II. 관련연구 .....</b>	<b>4</b>
1. SOA(Service Oriented Architecture) .....	4
2. 센서웹 .....	5
3. 지형지물 유일식별 .....	7
4. OPEN API 기반의 콘텐츠 전송 기술 .....	9
5. 프로그램 배포기술 .....	11
1) Smart Client .....	11
2) Click Once .....	13
<b>III. 센서웹 기반의 COST 설계 및 구현 .....</b>	<b>14</b>
1. 센서웹 모델 .....	14
1) 센서웹 모델 .....	14
2) 센서웹 시스템 구성 .....	18
2. 센서웹 구성요소 .....	21
1) 서비스 저장소 .....	21
(1) 서비스 저장소 주요기능 .....	21
(2) 서비스 저장소 설계 .....	22

(3) 서비스 저장소 구현 .....	24
2) 서비스 요청자 .....	28
(1) 서비스 요청자 주요기능 .....	29
(2) 서비스 요청자 설계 .....	30
(3) 서비스 요청자 구현 .....	32
3) 서비스 공급자 .....	36
(1) 서비스 공급자 주요기능 .....	36
(2) 서비스 공급자 설계 .....	37
(3) 서비스 공급자 구현 .....	43
3. COST 설계 및 구현 .....	47
1) COST 주요기능 .....	47
2) COST 설계 .....	48
3) COST 구현 .....	57
<b>IV. 성능분석 .....</b>	<b>64</b>
1. 실험 환경 .....	64
2. 실험 결과 .....	66
1) 콘텐츠 생성 및 전송 .....	67
2) 서비스 공급자 검색 및 콘텐츠 제공 .....	70
<b>V. 결론 .....</b>	<b>75</b>
<b>참고문헌 .....</b>	<b>76</b>

## 그림 목 차

그림 1. OGC Sensor Web Integrated Client .....	6
그림 2. UFID의 구성 .....	8
그림 3. WCF의 정의 .....	10
그림 4. WCF 아키텍처 .....	11
그림 5. Smart Client의 특징 .....	12
그림 6. Click Once 배포 .....	13
그림 7. 센서웹 모델 .....	14
그림 8. 센서웹과 COST .....	15
그림 9. 서비스 구현 예 .....	16
그림 10. 센서웹 시스템 세부 구성도 .....	18
그림 11. 센서웹과 COST 간의 시퀀스 다이어그램 .....	19
그림 12. 센서웹 OPEN API 정의 및 시스템 간 관계 .....	20
그림 13. 서비스 저장소 구성도 .....	22
그림 14. 서비스 저장소 시퀀스 다이어그램 .....	23
그림 15. 서비스 저장소 데이터베이스 객체 관계 다이어그램 .....	24
그림 16. 서비스 저장소의 클래스 다이어그램 1 .....	25
그림 17. 서비스 저장소의 클래스 다이어그램 2 .....	26
그림 18. 서비스 저장소의 실행과정 .....	27
그림 19. 서비스 등록 및 위치 서비스 정보 관리 실행화면 .....	28
그림 20. 전체 서비스 정보 관리 실행화면 .....	28
그림 21. 서비스 요청자 구성도 .....	30
그림 22. 서비스 요청자 시퀀스 다이어그램 .....	31
그림 23. 서비스 요청자의 클래스 다이어그램 1 .....	32
그림 24. 서비스 요청자의 클래스 다이어그램 2 .....	33
그림 25. 서비스 요청자의 실행과정 .....	34
그림 26. 서비스 검색 실행화면 .....	35



그림 27. 콘텐츠 뷰어 실행화면 .....	35
그림 28. 서비스 공급자 구성도 .....	37
그림 29. 서비스 공급자 시퀀스 다이어그램 .....	38
그림 30. 서비스 공급자 데이터베이스 객체 관계 다이어그램 .....	39
그림 31. 위도/경도와 화면 픽셀좌표의 매핑 순서도 .....	40
그림 32. 위치정보 매핑 설명 .....	41
그림 33. 서비스 공급자의 클래스 다이어그램 1 .....	43
그림 34. 서비스 공급자의 클래스 다이어그램 2 .....	44
그림 35. 서비스 공급자의 실행과정 .....	45
그림 36. 서비스 공급자 메인 실행화면 .....	46
그림 37. 센서정보 메모리 뷰어 실행화면 .....	46
그림 38. COST 구성도 .....	48
그림 39. 지도정보 생성 및 관리 모듈 .....	49
그림 40. 기타 정보 관리 모듈 .....	50
그림 41. 노드 정보 관리 모듈 .....	50
그림 42. 전체 서비스 콘텐츠 상태 뷰어 모듈 .....	51
그림 43. 서비스 공급자 정보 생성 및 관리 모듈 .....	51
그림 44. 서비스 콘텐츠 전송관리 모듈 .....	52
그림 45. COST 시퀀스 다이어그램 .....	53
그림 46. COST 데이터베이스 객체 관계 다이어그램 .....	54
그림 47. 지도 및 노드정보 생성 개요 .....	55
그림 48. 지도정보 생성 순서도 .....	56
그림 49. 노드정보 생성 순서도 .....	57
그림 50. COST의 클래스 다이어그램 1 .....	58
그림 51. COST의 클래스 다이어그램 2 .....	59
그림 52. COST의 실행과정 .....	60
그림 53. COST 실행화면 .....	61
그림 54. 지도정보 관리 실행화면 .....	62
그림 55. 기타정보 관리 실행화면 .....	63

그림 56. 서비스 제공자 정보 및 서비스 콘텐츠 전송 관리 실행화면 .....	63
그림 57. 실험환경 네트워크 구성 .....	65
그림 58. 가상 센싱 정보 생성과정 .....	66
그림 59. 가상 센싱 정보 실행화면 .....	66
그림 60. COST 실험 개념도 .....	67
그림 61. 콘텐츠 생성 .....	68
그림 62. 콘텐츠 생성의 소요시간 .....	69
그림 63. 콘텐츠 전송 .....	69
그림 64. 서비스 공급자 .....	70
그림 65. 콘텐츠 전송의 소요시간 .....	70
그림 66. 서비스 공급자 검색 .....	71
그림 67. 서비스 공급자 검색의 소요시간 .....	72
그림 68. 서비스 요청자 (공과대학 1층) .....	73
그림 69. 서비스 요청자 (공과대학 2층) .....	73
그림 70. 서비스 요청자 (공과대학 3층) .....	74
그림 71. 서비스 요청자 (공과대학 4층) .....	74
그림 72. 콘텐츠 제공의 소요시간 .....	75

## 표 목 차

표 1. 마이크로소프트 분산기술의 종류 .....	10
표 2. 서비스 저장소의 주요기능 .....	21
표 3. 서비스 요청자의 주요기능 .....	29
표 4. 서비스 공급자의 주요기능 .....	36
표 5. COST의 주요기능 .....	47
표 6. 실험 환경 .....	64

## 센서웹 기반의 COST 설계 및 구현

컴퓨터공학과 안연준

지도교수 김도현

최근 유비쿼터스 기술이 발달함에 따라 센서 네트워크에서 다양한 하드웨어를 통하여 수집한 상황정보를 도시하는 센서웹의 연구가 OGC(Open Geospatial Consortium)를 중심으로 진행된다. 센서웹은 인공물과 자연물에 컴퓨터 기능을 갖는 다양한 센서를 설치하고 이를 웹으로 모니터링 하는 개념이다. 이러한 센서웹은 SOA(Service Oriented Architecture)라는 개념을 기초로 하며 기본적인 아키텍처는 서비스 저장소(Service Registry), 서비스 요청자(Service Client), 서비스 공급자(Service Provider)로 구성된다.

센서웹에서 서비스 공급자는 매우 다양한 실시간 상황 정보를 제공하며 이를 관리하는 역할을 담당하며, 이를 위해서는 서비스 공급자별로 지도, POI(Point of Interest), 센서 노드, 구동체 노드 등을 포함하는 고유의 서비스 콘텐츠가 필요하며, 개별적으로 맞춤형 콘텐츠를 생성하고 유지관리하기 어렵다. 따라서 본 논문에서는 다양한 센서 네트워크로부터 실시간으로 수집되는 상황 데이터를 통합하고 처리한 정보를 제공하는 서비스 공급자를 위해 서비스 콘텐츠를 생성하고 관리하는 도구인 COST(COntents Supporting Tool)를 설계하고 구현한다. 이를 위해 기존의 센서웹 기술을 고찰하며, 센서웹과 COST를 연동하는 실험환경을 구축하여 검증한다.

제시된 센서웹 기반의 COST는 상황 데이터를 효율적으로 웹 서비스를 통해 제공할 수 있다. 이를 바탕으로 사용자의 요구에 따라 센서 네트워크의 상황 정보를 효율적으로 사용자에게 제공할 수 있을 것으로 사료된다. 또한 기존의 센서웹에 적용해 봄으로서, 향후 실시간 상황 정보를 수집하는 사용자에게 가시화하는 여러 시스템에도 적용 가능할 것으로 전망된다.

## ABSTRACT

# Design and Implementation of COST based on the Sensor Web

An, Yeon-Jun  
Department of Computer Engineering  
Graduate School  
Jeju National University

Recently, the OGC(Open Geospatial consortium)researches intothe Sensor Web visualized the situation information that collected through the varied sensing device in the sensor networkas the development of the ubiquitous technology.The S ensor Web is the concept that it establishes the varied sensor which has the many computingfunction in an artifact and a natural object, and monitors them using the web. Such Sensor Web is based on the concept called SOA(Service Oriented Architecture) and this architecture consists of the Service Registry, Service Client, Service Provider.

In the Sensor Web, the Service Provider provides and manages so varied real-time information, and each Service Provider needs the unique service contents which included the map, POI, sensor node, actuator node, etc. But it is so difficult to create and manage the customized contents individually. Accordingly, in this paper, we design and implement the COST(Contents Supporting Tool) which creates and manage the service contents for the Service Provider which supports the collectedreal-time context data from the varied sensor networkuser. Torealize this, weanalysis the technology of Sensor Web.

And we implement and verify the experimental environment interworking with Sensor Web and COST.

The proposed the COST based on the Sensor Web provide economically the context knowledge through the web service. Based on this, I think it can provide the context information in the sensor network according to the request of the user with efficiency. And, in future, the proposed system is expected to use for visualizing many other systems collected the real time sensor information as applying it to the existed Sensor Web.

## 약어표

COST	COntents Supporting Tool
OGC	Open Geospatial Consortium
SWE	Sensor Web Enablement
SOA	Service Oriented Architecture
W3C	World Wide Web Consortium
UDDI	Universal Description Discovery and Integration
SPS	Sensor Planning Service
WMS	Web Mapping Service
SOS	Sensor Observations Service
SAS	Sensor Alert Service
SensorML	Sensor Model Language
O&M	Observations & Measurements
TML	Transducer Model Language
UFID	Unified Feature Identification
TOID	Topological Identifiers
ASDI	Australian Spatial Data Infrastructure
OOP	Object-Oriented-Programming
CBD	Component-Based-Develop
WSE	Web Services Enhancements
MSMQ	Microsoft Message Queuing
WCF	Windows Communication Foundation
SOAP	Simple Object Access Protocol
POI	Point of Interest
URI	Uniform Resource Identifier
WSDL	Web Service Description Language



## I. 서론

### 1. 연구 배경

최근 국제적으로 인터넷 웹 상에서 다양한 상황 데이터를 제공하는 센서 웹에 대한 연구가 OGC(Open Geospatial Consortium)를 중심으로 미국의 마이크로소프트, 미국방성, 호주의 멜버른 대학교 등에서 진행되고 있다. 센서웹은 인공물과 자연물에 컴퓨터 기능을 갖는 다양한 센서를 설치하고 이를 웹으로 연결시켜 시설물 및 환경, 교통 상태, 재난재해 등을 모니터링 하는 개념이다. 또한, 향후에는 이러한 센서웹을 도시 전체에 연결하여 u-City를 구축하거나 더 나아가 지구적인 네트워크로 연결하여 전자 센서망을 가진 지능적인 지구 공간 환경으로 발전할 것이다[1][2].

OGC는 지도 관련 국제 표준 기구로 센서웹 그룹에서는 SWE(Sensor Web Enablement) 표준화를 진행하고 있다. SWE 프레임워크 내의 서비스간 전형적인 협업 구조를 보여주고 있다. 센서웹은 정확하고 안정된 데이터 수집을 통해 실시간 탐지 및 조기 경고 시스템의 구축이 가능하도록 한다. 이는 산불, 지진, 해일 등 위기 상황에 있어서 즉각적인 대비가 가능하게 한다[3][4].

이러한 센서웹은 SOA(Service Oriented Architecture)라는 개념을 기초로 하고 있다. SOA란 서비스로 구성되어 있으며 서비스의 정의는 공통된 인터페이스를 통해 자신이 가진 비즈니스 프로세스를 처리할 수 있는 독립적으로 구성된 컴포넌트를 의미한다. SOA의 기본적인 아키텍처는 서비스 저장소(Service Registry), 서비스 요청자(Service Client), 서비스 공급자(Service Provider)로 구성된다[6][7].

본 논문에서는 다양한 센서 네트워크로부터 실시간으로 수집되는 상황 데이터를 통합하고 처리한 정보를 제공하는 서비스 공급자를 위해 COST(COntents Supporting Tool)를 설계 및 구현한다. 이를 위해 기존의 센서웹 기술을 고찰하며, COST를 설계 및 구현하고 센서웹과 COST를 연동하는 실험환경을 구축하여 검증한다.

## 2. 연구목적 및 방법

센서웹은 인공물과 자연물에 컴퓨터 기능을 갖는 다양한 센서를 설치하고 이를 웹으로 연결시켜 시설물 및 환경, 교통 상태, 재난재해 등을 모니터링 하는 개념이다. 이러한 센서웹은 SOA라는 개념을 기초로 하며 기본적인 아키텍처는 서비스 저장소, 서비스 요청자, 서비스 공급자로 구성된다. 센서웹에서 서비스 공급자는 다양한 센서 네트워크로 부터 실시간으로 수집되는 상황 데이터를 통합하고 처리한 정보를 서비스 요청자에게 제공하는 역할을 수행한다. 서비스 공급자는 매우 다양한 실시간 상황 정보를 제공하며 이를 관리하는 역할을 담당하며, 이를 위해서는 서비스 공급자별로 지도, POI(Point of Interest), 센서 노드, 구동체 노드 등을 포함하는 고유의 서비스 콘텐츠가 필요하며, 개별적으로 맞춤형 콘텐츠를 생성하고 유지관리하기 어렵다.

따라서 본 논문에서는 센서웹을 지원하는 COST를 설계하고 구현한다. 다양한 센서 네트워크로부터 실시간으로 수집되는 상황 데이터를 통합하고 처리한 정보를 제공하는 서비스 공급자를 위해 서비스 콘텐츠를 생성하고 관리하는 도구인 COST를 제시하였다. 이를 위해 기존의 센서웹 기술을 고찰하며, 센서웹과 COST를 연동하는 실험환경을 구축하여 검증한다. COST의 주요기능은 서비스 콘텐츠 상태의 모니터링과 지도정보, POI, 고정형/이동형 센서 및 구동체정보, 위치매핑정보, 서비스 제공자정보를 생성 및 관리하고 서비스 콘텐츠 전송을 담당하며 서비스 공급자를 위한 프로그램을 배포한다.

제시된 센서웹 기반의 COST는 상황 데이터를 효율적으로 웹 서비스를 통해 제공할 수 있다. 이를 바탕으로 사용자의 요구에 따라 센서 네트워크의 상황 정보를 효율적으로 사용자에게 제공할 수 있을 것으로 사료된다. 또한 기존의 센서웹에 적용해 봄으로서, 향후 실시간 상황 정보를 수집하는 사용자에게 가시화하는 여러 시스템에도 적용 가능할 것으로 전망된다.



### 3. 논문 구성

서론에 이어 2장 관련 연구에서는 SOA의 분석과 기본적인 아키텍처를 고찰하고, 센서웹에 대한 분석과 해외사례 조사 및 관련기술 OPEN API 기반의 콘텐츠 전송 기술을 조사하고 프로그램 배포기술을 분석한다. 3장에서는 센서웹 모델을 소개하고 센서웹을 구성하는 서비스 등록소, 서비스 요청자, 서비스 제공자를 설명하며 본 논문에서 제시하는 센서웹 기반의 COST의 주요기능 및 설계, 구현결과를 설명하고 제안하였다. 4장에서는 3장에서 기술한 설계 및 구현을 토대로 서비스 콘텐츠 생성 및 전송에 대해 성능분석을 하였다. 마지막으로 5장에서는 결론을 맺는다.

## II. 관련연구

### 1. SOA(Service Oriented Architecture)

가트너(Gartner) 그룹이 정의한 SOA는 여러 개의 등록된 서비스들과 서비스 소비자(클라이언트)들 간의 느슨한 연결 관계로 구성된 소프트웨어 구조라고 하며 서비스는 외부의 컨텍스트가 정의된 프로그램 인터페이스를 통해 호출 가능하게 설계된 비즈니스 컴포넌트라고 한다. 또한 W3C(World Wide Web Consortium)는 SOA를 호출 가능한 컴포넌트의 집합이며 컴포넌트는 인터페이스 정의가 공개되고 발견 가능한 것이라고 정의하고 있다. 따라서, SOA는 재사용 가능한 서비스의 형태로 통합 IT 컴퓨팅 자산을 구성하는 아키텍처이고, 비즈니스 변화에 즉각적인 효과를 발휘할 수 있는 서비스 관점의 소프트웨어 아키텍처라고 할 수 있다. 또한 SOA에서의 서비스는 아래의 추가적인 요구사항이 필요하다.

- 플랫폼에 독립적 : SOA에서의 서비스는 표준화된 방법을 통해 모든 환경에서 호출이 가능해야 한다. 이는 서비스 호출 메커니즘이 널리 채택된 표준에 근거해야 함을 의미한다.
- 약결합 방식 : 서비스는 그 이용에 있어 내부 자료 구조나 지식을 필요로 해서 안된다.
- 위치 투명성의 지원 : 서비스는 그들의 정의와 위치정보를 UDDI(Universal Description Discovery and Integration)와 같은 저장소에 저장하고 여러 클라이언트를 통해 그들의 위치와 상관없이 등록, 호출될 수 있어야 한다.

즉, SOA란 서비스라 불리는 분할된 애플리케이션 조각들을 단위로 약결합으로 연결하여 하나의 완성된 애플리케이션을 개발하기 위한 소프트웨어 아키텍처라 하더라도, 사용자 어플리케이션, 또는 기타 서비스 어플리케이션의 기능을 서비스 형식

으로 전달할 수 있는 분산 시스템으로 구축하는 접근방법이라 말할 수 있다. 여기서의 서비스는 독립적인 비즈니스 기능을 구현한 소프트웨어 컴포넌트로서, 어플리케이션이나 다른 서비스가 외부에서 그 서비스를 찾을 수 있고, 공개된 인터페이스를 통해 접근하며, 주로 메시지 기반의 비동기 커뮤니케이션 방식으로 사용하도록 하는 것이다.

SOA의 구성요소는 서비스 저장소, 서비스 요청자, 서비스 공급자로 나눌 수 있다. 각각의 역할을 정리하면 서비스 저장소는 서비스에 대한 기술 정보(Description)를 저장 및 검색하는 역할을 하고 서비스 요청자는 서비스 공급자에 의해 제공되는 하나 이상의 서비스를 이용하며 서비스 공급자는 서비스 요청자가 호출 시 입력하는 값을 가공하여 해당하는 결과를 제공한다.

이러한 구조를 통해 SOA는 서비스의 발견과 동적 바인딩이라는 개념을 지원한다. 또한 서비스는 독립적으로 개발, 유지, 관리되며, 서로의 작동 자체에 큰 영향을 미치지 않는 독립적인 모듈의 개념이다[6][7].

## 2. 센서웹

센서 웹은 기존의 센서, 카메라, 측정 도구 등과 같은 상황 정보를 수집하는 장치와, 이들 장치로부터 수집되는 센싱 데이터를 저장하는 레포지토리(repository)를 웹 상에서 발견하고 접근하거나 제어할 수 있도록 지원한다. 센서 웹은 수많은 센싱 데이터를 저장하고 관리하며, 사용자들에게 인터넷 상으로 제공하는 것을 목적으로 하고 있다.

현재 OGC web service, phase-4(OWS-4)에서는 센서웹을 SWE라는 이름의 개방형 플랫폼으로 표준화가 거의 완료 단계에 있으며, 이는 웹을 기반으로 모든 센서를 발견하고, 센서를 통해 데이터 획득 및 교환(acquisition & exchange), 정보처리, 임무부여(tasking) 등을 수행할 수 있도록 한다.

세부적인 표준화 사양을 살펴보면 SPS(Sensor Planning Service), WMS(Web Mapping Service), Registry, SOS(Sensor Observations Service), SAS(Sensor Alert S

ervice), SensorML(Sensor Model Language), O&M(Observations & Measurements) Schema, TML(Transducer Model Language), 등으로 구성되며 이를 각각 구현함으로써 센서웹을 실현할 수 있도록 하고 있다[5].

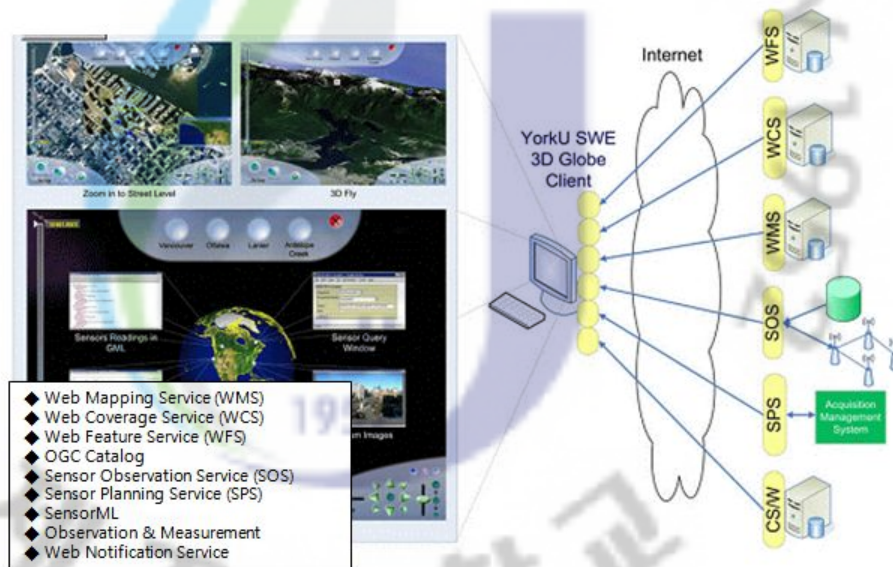


그림 1. OGC Sensor Web Integrated Client

SPS는 사용자가 웹을 통해 연결되어 있는 센서에 임의의 임무를 부여하고 이를 수행한다. SPS는 사용자로부터 센서가 수행할 임무에 대한 인자 값을 전달받는 기능, 사용자가 요청한 임무가 실행 가능한지 그 여부를 알려주는 기능, 해당 임무를 실제 수행하는 기능 등을 지원한다.

SOS는 센서 또는 센서 시스템으로부터 관측된 데이터에 대한 접근을 제공하는 표준 인터페이스로서, 센서를 사용하는 사용자들 사이에 발생할 수 있는 용어 및 관점의 차이를 제거하는 것을 지원한다. SOS에서는 SensorML과 O&M을 사용하여 센서와 센싱 정보에 접근하고 처리한다.

Sensor web registry는 전체 센서웹 프레임워크 상에서 앞의 모든 모델링 및 서비스들을 검색하고 발견하기 위한 표준 디렉토리 서비스로서, 센서 정보 또는 센서의 함수 정보, 센싱 데이터 정보, 태스크 정보, 경보 정보 등을 검색 및 추가, 삭제, 갱신하는 기능을 제공한다.

SAS는 온도, 습도, 조도 등의 센서에서 센싱된 데이터가 특정 한계치를 넘는 경우나 특정한 상황이 발생된 경우, 또는 센서의 상태 정보가 변경된 경우 등을 이벤트로 정의하고, 해당 이벤트에 대한 경보 메시지를 사용자에게 전달하는 표준 인터페이스이다. 이를 위해서 SAS는 사전에 정의된 경보 가능한 메시지를 registry에 등록하고, 사용자는 registry 중에서 관심 있는 서비스에 가입함으로써 경보 메시지를 수신할 수 있다.

O&M은 센서가 관측 또는 측정된 센싱 정보를 인코딩하는 XML 기반의 표준 모델로서, 특정 센서 또는 특정 단체에 종속되는 데이터 포맷으로만 해석되는 문제를 없애기 위함이다.

WNS는 SAS가 사용자에게 이메일, SMS, HTTP, 전화, 팩스 등을 통해 전달되도록 하는 표준 인터페이스로서, HTTP의 request/response와 같은 동기적인 알림 처리뿐만 아니라 비동기적인 알림도 지원한다.

SensorML은 온도, 습도, 조도 등과 같은 현장센서(in-situ sensor)에서 웹캠, CCTV, 위성영상센서, 항공영상센서와 같은 원격센서(remote sensor)에 이르기까지 모든 다양한 센서들을 추상화하기 위한 XML 기반의 표준 모델이다. 즉, 센서에 대한 사전 지식이 없이도 센서를 발견하고 센싱 정보를 해석할 수 있도록 센서에 대한 메타정보를 제공한다.

TML은 센서와 구동장치를 합한 트랜스듀서에 관한 정보를 모델링하는 함수와 메시지 포맷으로서, 트랜스듀서의 데이터를 획득하고 저장 및 전달하는 공통 포맷을 제공한다.

이상에서 언급한 SWE는 현재 OGC에서 테스트베드를 구축하여 컴포넌트 구현 및 테스트를 수행 중에 있으며, 이미 표준화된 WFS, WMS, WCS 등 과도 상호 연동되도록 함으로써 모든 공간정보의 통합 및 융합, 공유 측면에서 큰 중요성을 갖는다[1].

### 3. 지형지물 유일식별

유비쿼터스 세상의 핵심은 공간상에 존재하는 모든 물체를 고유하게 식별하고 쌍



방향통신을 통해 하나로 연결하는 것이다. 유비쿼터스 환경에서는 공간상의 대상물이 정확하게 좌표계로 표현되고 이 안에서 모든 사물의 움직임이 실시간으로 측정된다. 결국 유비쿼터스 컴퓨팅은 현실세계의 모든 대상물을 고유하게 식별하기 위한 체계와 이를 컴퓨팅 환경 하에서 실시간으로 관리할 수 있는 기술을 기반으로 구현된다.

유비쿼터스 컴퓨팅을 위한 주요기술 중 하나인 UFID(Unified Feature Identification)는 지형지물의 유일 식별자로 지리적 위치와 관련이 있는 객체에 부여하는 식별기호를 갖는다. 객체를 지정하는 유일 식별자로써의 UFID는 대상 객체의 생애 주기에 따라 생성, 소멸하며 수정이 가능하다. 지형지물 전자식별자(UFID)는 개별적으로 관리되는 지형지물에 대해 위치정보, 도로, 건물, 하천 등의 지형지물 종류, 지형지물 관리기관, 기타 속성정보 등을 나타내는 유일한 단일식별자를 부여함으로써 도로, 건물 등의 인공적 지형지물과 하천등의 자연적 지형지물을 체계적으로 관리 및 활용할 수 있도록 해준다. 따라서 UFID는 주민등록번호처럼 우리나라의 국토를 구성하고 있는 도로, 건물 및 하천 등의 모든 인공적 및 자연적 지형지물에 단일식별자를 부여함으로써 해당 지형지물을 관리하는 기관은 물론 물류, 금융 등 각종 산업분야에 매우 중요한 역할을 하게 된다.

UFID (Unique Feature Identifier)							
버전	지형지물	관리기관	일련번호	위치정보	고도정보	속성	오류확인
2 byte	4 byte	6 byte	3 byte	11 byte	2 byte	1 byte	1 byte

그림 2. UFID의 구성

UFID는 버전, 지형지물, 관리기관, 일련번호, 위치정보, 고도정보, 속성, 오류확인으로 구성된다. 버전은 UFID임을 확인하며 버전의 수, 지형지물은 국토지리정보원의 지형지물 분류체계, 관리기관은 지형지물의 관리 담당 기관, 일련번호는 일정구역 내부의 동일 지형지물 분류, 위치정보는 지형지물의 위치정보, 고도정보는 지형지물의 고도정보, 속성은 지형지물의 다른 정보 유무, 오류확인 ID의 전송오류 확인을 나타낸다[9][10].

최근 국외 현황 중 미국의 National Map는 여러 가지 데이터 합성기술의 개발과 영구 Feature ID를 위한 대표 ID의 정의 및 사용방법 등에 대한 연구, 데이터의 통합을 위한 전략 등을 연구하고 있다. 영국의 Master Map은 실세계를 반영하는 연속적인 데이터베이스로써 유일식별자인 TOID(Topological Identifiers)를 이용하여 다양하게 데이터를 활용할 수 있는 특징을 가지고 있다. 호주의 ASDI(Australian Spatial Data Infrastructure)는 조화로운 데이터 구조(HDF, Harmonised Data Framework)를 주 목적으로 영구식별자(PID)를 부여하는 방법 및 내용에 대한 기술을 Feature에 대한 유일자 성격을 가지고 유지 및 관리하기 위한 방법으로 도입하고 있다. 국내에서는 국토지리정보원(구 국립지리원)의 수치데이터모델연구와 지형지물전자식별자 활용기술연구 등 주요한 UFID 관련 연구들이 수행되어 왔다. 이 연구들에서는 다양한 UFID 체계의 구성요소들이 발굴 되었다.

#### 4. OPEN API 기반의 콘텐츠 전송 기술

2000년도에 이르러서 SOA(Service-Oriented-Architecture) 기술이 큰 유행이 불게 되었다. OOP(Object-Oriented-Programming)나 CBD(Component-Based-Develop)와 같은 개념들은 과거에 굉장히 유행했던 아키텍처의 패턴 이었다. 2000년도에서도 역시 새롭게 등장한 패턴이 바로 SOA라는 서비스 지향적 아키텍처이다. 보편적으로 SOA는 상호 관계를 가지고 있는 기업들간의 정보 시스템을 구현할 때 많이 사용된다. SOA를 구축하기 위해서는 기업이나 고객간의 분산 환경을 같이 상호 운용이 가능하게 설계 되어야 한다. 이때 일반적으로 기업끼리는 분산환경에 대해 서로 협의를 하게 된다. TCP 통신을 이용할 것인지, SOAP(Simple Object Access Protocol) 통신을 이용할 것인지, 닷넷끼리의 환경이라면 Remoting 서비스도 대두될 것이다. 서비스 아키텍처를 협의에서 오르내리게 될 마이크로소프트의 분산기술은 다음과 같이 구성되어 있다.

서비스 이름	기술설명
웹 서비스	상호 운용 가능한 일반적인 Web 서비스(Basic Profile 1.1 준거)를 구축하기 위한 테크놀로지
WSE (Web Services Enhancements)	WS-*로 불리는 Web 서비스의 최신의 확장 사양을 가능한 서포트한, 상호 운용을 의식한 차세대 Web 서비스 구축을 위한 테크놀로지
.NET Remoting	객체의 위치 투과성을 공급하고, 리모트 콜과 로컬 콜의 차이를 은폐한 프로그래밍을 가능하게 하는 분산 객체를 구축하기 위한 테크놀로지
Enterprise Services	분산 트랜잭션이나 객체 풀링 등 컴포넌트의 작성과 사용을 단순화 하면서, 어플리케이션의 범위성과 융통성을 향상시키기 위한 고도의 기능을 제공하는 테크놀로지
MSMQ (Microsoft Message Queuing)	즉시 응답을 필요로 하지 않는 비 동기 메시지 교환의 어플리케이션을 구축하기 위한 테크놀로지

표 1. 마이크로소프트 분산기술의 종류

이런 마이크로소프트의 분산기술을 소개한 이유는 위의 기술들이 바로 WCF(Windows Communication Foundation)와 연관이 있기 때문이다. WCF는 마이크로소프트의 분산 기술을 통합하자는 의미에서 새롭게 탄생되었다.

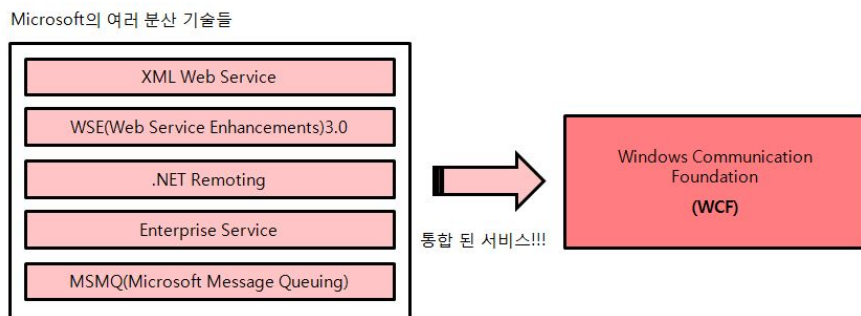


그림 3. WCF의 정의

WCF는 서비스 지향 어플리케이션을 신속하게 개발하기 위한 통합된 서비스 프로그래밍이라고 정의할 수 있다[8].



.NET Framework 3.0

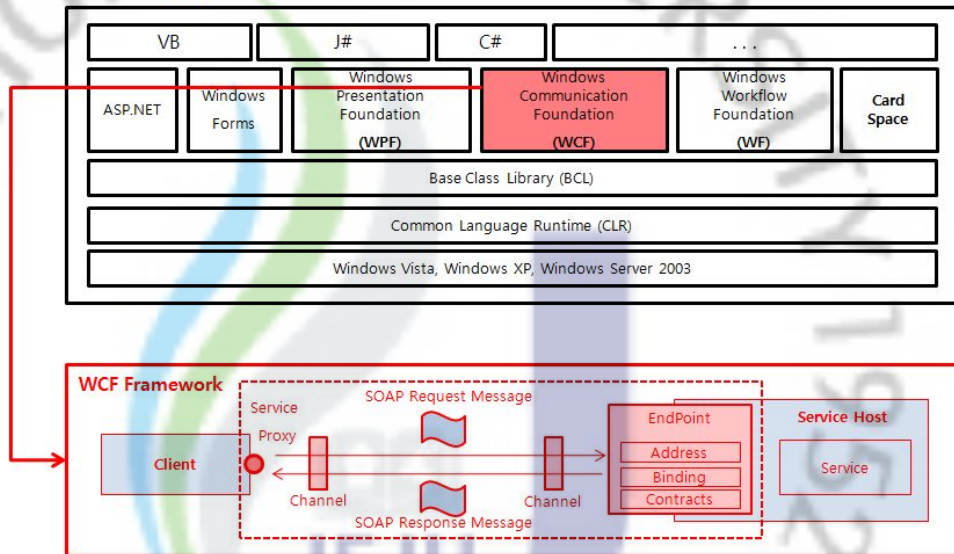


그림 4. WCF 아키텍처

WCF는 .Net Framework 3.0에 새롭게 추가된 기술의 하나이다. WCF에서 서비스와 클라이언트는 EndPoint를 통해서 통신을 하게 된다. 서비스에서는 여러 개의 EndPoint가 존재 할 수 있으며, 클라이언트는 EndPoint를 통해서 메시지를 주고 받게 된다. Endpoint는 Address, Binding, Contract로 구성되며 어디에서 무엇을 어떻게 보낼 것인지에 대한 규약을 나타낸다. 클라이언트는 Service Proxy를 통해 EndPoint와 접속을 한다. Channel은 EndPoint와 메시지를 보내고 받기 위한 핵심 추상이고 Transport Channel과 Protocol Channel로 구분된다. Transport Channel은 TCP, UDP, MSMQ등의 전송 프로토콜을 사용해 스트림을 보내거나 받는 작업을 처리한다. Protocol Channel은 메시지를 처리하고 수정하는 SOAP 기반 프로토콜을 구현한다.

## 5. 프로그램 배포기술

### 1) Smart Client

1990년대 중반부터 인터넷 열풍으로 소프트웨어 산업의 대대적인 개편이 일어나고 인터넷 네트워킹 분야에 대한 웹 개발이 증가된다. 웹 기반 어플리케이션은 썬 클라이언트(Thin Clients)라고도 일컬어지는데 웹 브라우저를 통해 장소의 한계를 극복하여 배포의 간편화 및 업그레이드에 대한 비용 절감의 특성을 갖고 있다. 과거의 썬 클라이언트는 사용자 인터페이스 제공의 필요성이 대두되고 추가 기능이 요구된다. 또한 이식성과 표준 문제가 있으며 수행 효율에 대한 기술적인 한계가 존재 한다. 이로 인해 리치 클라이언트(Rich Client : Thick 및 Fat Client 라고도 불려진다.)의 필요성이 대두 되었다.

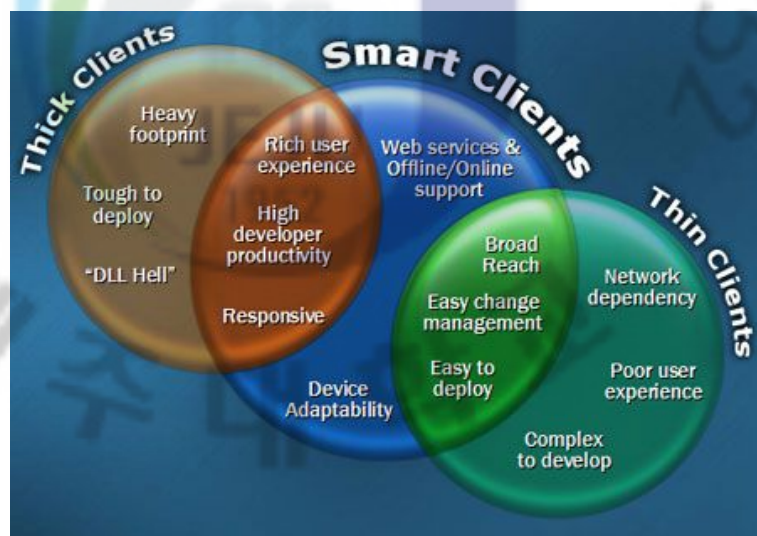


그림 5. Smart Client의 특징

Smart Client는 클라이언트를 위한 다양한 기술 제공하며, 개발 시 ActiveX 보다 쉽게 개발이 가능하다. 이로 인해 개발 생산성 증가 효과, 제작 언어에 대한 한계 탈피, 인증에 대한 내용 강화의 장점을 갖게 되며, 단점으로 프레임워크에 의존적인 문제점 갖는다. 코드 액세스 보안을 통해 사용자에게 실행되는 모든 코드에 코드액세스 보안 적용되며, 악의적인 코드의 실행 방지 및 확산을 줄 수 있다. 또 다른 특성중 하나로 배포의 용이함이 있고, 통합된 자체 버전 관리로 'DLL 지옥'문제 해결과, 버전에 대한 통합 적인 관리 체계 및 자동화 기능을 지원한다. 추가적으로 Smart Client는 배포 시 온라인 및 오프라인 기능을 모두 제공한다[11].

개발한 COST에서는 사용자에게 지도 상에서 웹 기반의 실시간 상황 정보를 제공하기 위

해 Smart Client 배포 기술을 사용한다.

## 2) Click Once

Click Once는 Windows Forms 기반 응용 프로그램을 배포할 수 있게 해주는 새로운 응용 프로그램 배포 기술이다. Click Once를 사용할 경우 웹 페이지에서 링크를 클릭하는 것과 같이 간단하게 Windows Forms 응용 프로그램을 실행할 수 있다. 또한 관리자의 경우에 응용 프로그램을 배포하거나 갱신하려면 단순히 서버의 파일을 업데이트하면 되므로 모든 클라이언트를 개별적으로 처리할 필요가 없다.

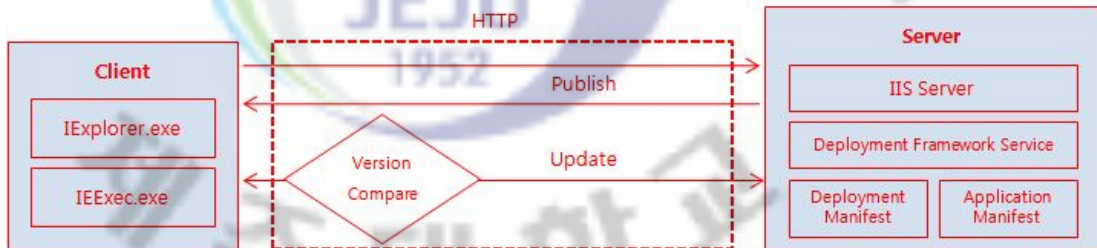


그림 6. Click Once 배포

핵심 Click Once 배포 아키텍처는 두개의 XML 매니페스트 파일인 응용 프로그램 매니페스트 및 배포 매니페스트를 기반으로 한다. 응용프로그램 매니페스트는 어셈블리, 응용 프로그램을 구성하는 종속성과 파일, 필수 권한 및 업데이트를 사용할 수 있는 위치를 포함하여 응용 프로그램 자체에 대해 설명하는 역할을 한다. 배포 매니페스트는 응용 프로그램 매니페스트의 위치 및 클라이언트에서 실행해야 되는 응용 프로그램 버전을 포함하여 응용 프로그램 배포 방법에 대해 설명하는 역할을 한다. 클라이언트는 HTTP 기반으로 웹 페이지나 폴더에서 배포 매니페스트 파일을 나타내는 아이콘을 클릭하여 응용 프로그램을 다운로드하고 설치할 수 있다. 또한 어셈블리의 버전을 비교하여 자동으로 응용 프로그램의 버전을 관리한다[12].

본 논문에서 Click Once 기술은 서비스 공급자에게 필요한 운영 프로그램을 배포하기 위해 사용한다.

### III. 센서웹 기반의 COST 설계 및 구현

본 논문에서는 센서 네트워크로 부터 실시간으로 수집되는 상황 데이터를 통합하고 처리한 정보를 제공하는 서비스 공급자를 위해 COST를 제시한다. COST는 서비스 콘텐츠 상태의 모니터링과 지도, POI, 고정형/이동형 센서 및 구동체, 위치매핑, 서비스 공급자 정보를 생성 및 관리하고 서비스 공급자에게 콘텐츠를 전송한다. 이를위해 관련 연구의 그림 7과 같은 SOA의 기본적인 아키텍처를 구현한 센서웹에서 COST를 구현하고 센서웹에서 서비스 저장소와 서비스 공급자와 연동 및 성능을 분석한다.

#### 1. 센서웹 모델

##### 1) 웹서버 모델

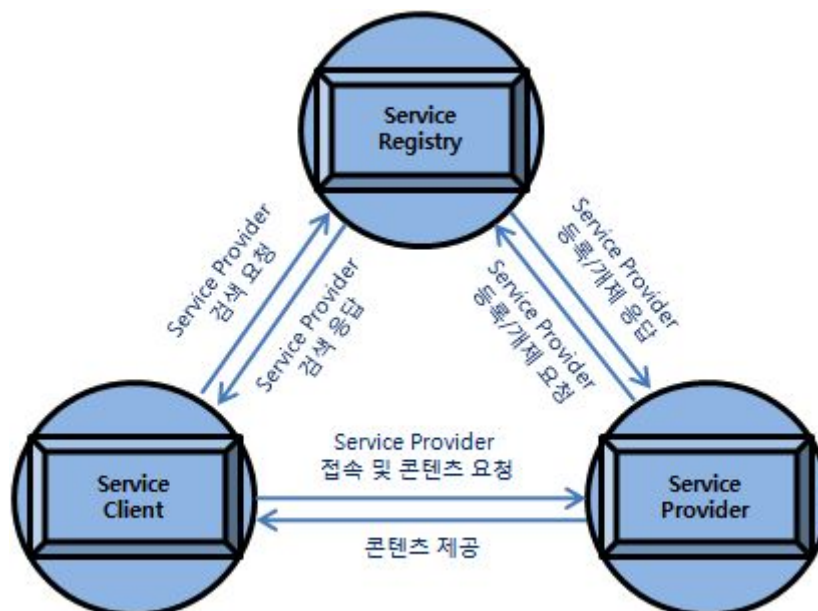


그림 7. 센서웹 모델



그림 7은 센서웹 모델을 나타낸다. SOA 기반의 센서웹은 서비스 저장소, 서비스 요청자, 서비스 공급자로 구성된다. 서비스 요청자는 서비스 저장소에 서비스 공급자를 검색하고 검색결과를 응답받는다. 검색결과는 공급자의 접속주소 및 WSDL 제공여부, 기타정보이고, 서비스 요청자는 해당 서비스 공급자로 접속하여 콘텐츠를 제공받는다. 서비스 공급자는 서비스 저장소에 자신의 공급자 정보를 등록하고 처리결과를 응답받는다.

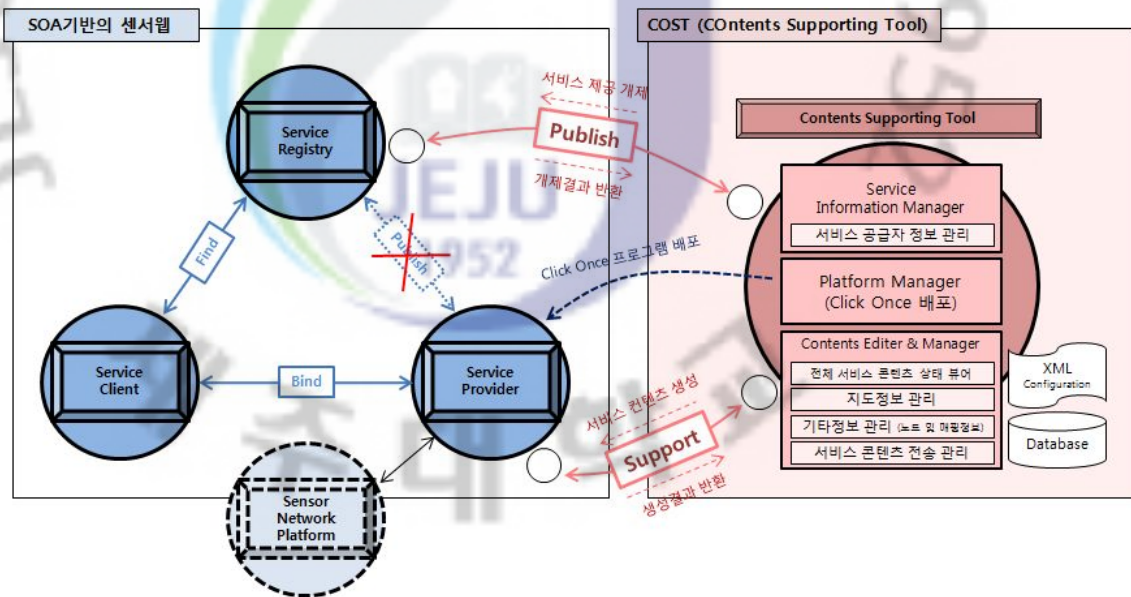


그림 8. 센서웹과 COST

본 논문에서 제안하는 COST는 SOA 기반의 센서웹에서 콘텐츠와 서비스 공급자를 통합 지원하는 기능을 지원한다. 그림 8에서 COST는 Contents Editor & Manager, Platform Manager, Service Information Manager로 구성된다. Contents Editor & Manager는 지도, 노드(POI, 센서, 구동체)와 같은 콘텐츠 정보를 생성하고 관리하며 서비스 공급자로 콘텐츠를 전송하는 역할을 담당한다. Platform Manager는 Click Once 배포 기술을 통해 서비스 공급자에서 동작하는 프로그램을 생성하고, 배포하는 역할을 담당한다. Service Information Manager는 서비스 공급자의 정보를 일괄적으로 관리하고 서비스 저장소에 서비스 공급자의 정보를 등록하는 역할을 담당한다.

실제 센서웹의 서비스 구현의 예는 그림 9와 같다. 제주대학교, 제주 한라대학교는 서비스 공급자, 서비스 검색 관리센터는 서비스 저장소, 사용자는 서비스 요청자를 나타내고, 프로그램 제공서버는 소프트웨어의 배포를 위한 역할을 나타낸다. 콘텐츠 관리자는 본 논문에서 제안하는 COST를 나타낸다. 각각의 서비스 공급자는 외부 센서 네트워크와 연동하여 실시간 센싱 데이터를 제공받는다. 서비스 공급자는 외부 시스템과 연동하여 실시간 센싱정보를 제공받고 각 대학의 지도 정보 및 노드(지도/위치매핑/센서/구동체/POI정보)를 서비스 요청자에게 제공한다. 서비스 요청자는 해당하는 서비스 공급자에 접속하기 위해 서비스 저장소에게 공급자 정보를 검색한다. 콘텐츠 지원도구는 SOA 기반의 센서웹에서 서비스 공급자에 콘텐츠를 생성하고 전송 및 관리하기 위한 기능을 지원한다. 전체 서비스에서 모든 통신 및 서비스 제공은 WCF 기반으로 OPEN API를 통해 제공된다. 서비스 공급자의 프로그램은 Click Once와 서비스 요청자는 Smart Client를 통해 배포된다.

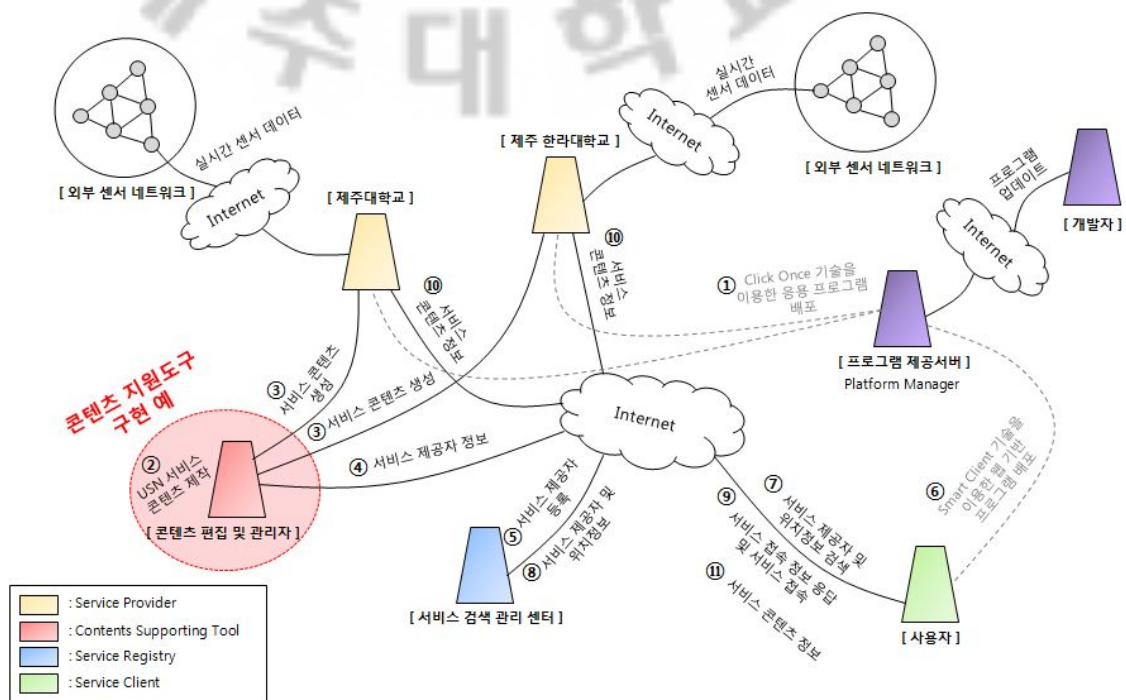


그림 9. 서비스 구현 예

세부적인 센서웹에서 서비스 구현 예의 흐름은 다음과 같다.

- 서비스 공급자는 ①에서와 같이 기본적으로 프로그램 제공서버의 Click Once 배포기술을 이용해 서비스 제공을 위해 서비스 제공자의 운영 프로그램이 설치된다.
- 콘텐츠 편집 및 관리자는 ②에서와 같이 USN 서비스 콘텐츠(지도/위치매핑/센서/구동체/POI정보)를 생성하고 관리한다.
- 콘텐츠 편집 및 관리자는 ③에서와 같이 제주대학교, 제주 한라대학교와 같은 서비스 공급자에게 서비스 콘텐츠를 등록하고, ④⑤에서 콘텐츠 관리자는 서비스 저장소에게 각각의 서비스 공급자의 정보를 등록한다.
- 서비스 요청자는 ⑥에서와 같이 프로그램 제공서버에서 Smart Client 배포 기술을 통해 웹 브라우저에서 프로그램을 실행한다.
- 서비스 요청자는 ⑦⑧에서와 같이 원하는 서비스 공급자를 검색하고 해당하는 공급자의 주소 및 접속정보를 획득한다.
- 이후 ⑨⑩⑪에서 서비스 요청자는 해당 서비스 공급자에게 접속하고 해당 서비스 콘텐츠를 제공 받는다.

## 2) 센서웹 시스템 구성

그림 10은 센서웹 시스템의 세부 구성도를 나타낸다. 서비스 저장소는 서비스 구동정보를 모니터링하고 서비스의 구동을 제어하고 전체 서비스 정보를 관리하며 위치 서비스 정보를 관리한다.

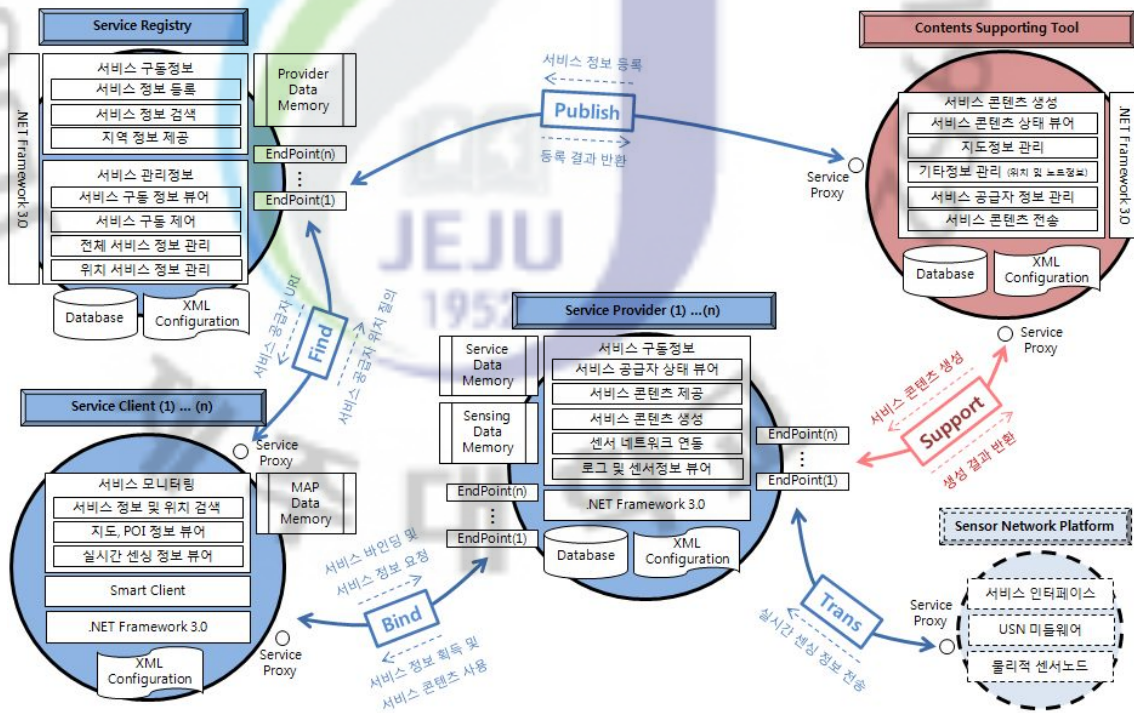


그림 10. 센서웹 시스템 세부 구성도

서비스 요청자는 서비스 정보 및 위치를 검색하고 콘텐츠 정보(지도, POI, 센서, 구동체, 실시간 센싱정보)를 모니터링 한다. 서비스 공급자는 서비스 콘텐츠 제공 및 생성, 센서 네트워크 연동, 로그 및 센서정보/서비스 공급자 상태를 모니터링 한다. 센서 네트워크 플랫폼은 센서 네트워크에서 수집되는 실시간 상황 데이터를 생성하고 서비스 공급자와 연동하여 실시간 정보를 제공하는 역할을 한다. 본 논문에서 제안하는 COST는 서비스 콘텐츠 상태 모니터링, 지도정보 및 기타정보(위치, 노드정보), 서비스 공급자 정보를 관리하고 콘텐츠 전송의 역할을 한다. COST는 서비스 공급자의 정보를 등록하고, 서비스 공급자에게 콘텐츠를 생성한다.



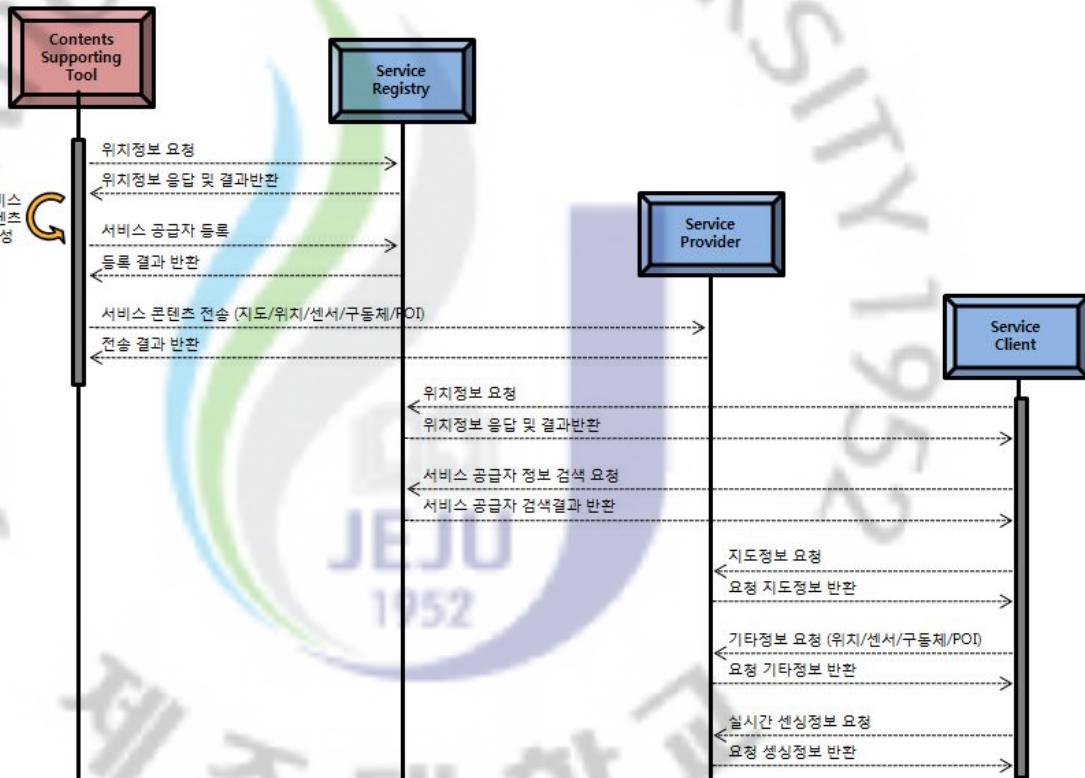


그림 11. 센서웹과 COST 간의 시퀀스 다이어그램

그림 11은 센서웹과 COST 간의 시퀀스 다이어그램을 나타낸다. 전체 구조는 COST, 서비스 저장소, 서비스 공급자, 서비스 요청자로 구성된다. COST에서는 서비스 저장소로 위치정보를 요청하고 결과를 응답 받는다. 응답받은 위치정보를 기초로 서비스 공급자의 정보를 생성 및 관리하고 또한 지도, 센서, 구동체, POI와 같은 서비스 콘텐츠를 생성한다. 서비스 공급자 정보는 서비스 저장소에 등록되며, 서비스 콘텐츠 정보는 서비스 공급자에게 서비스 콘텐츠를 등록하고 결과를 응답받는다. 서비스 요청자는 서비스 저장소에게 위치정보를 요청하고 응답받고, 위치정보와 서비스 공급자 검색정보를 통해 서비스 저장소에서 서비스 공급자를 검색하고 접속 정보를 응답받는다. 서비스 공급자 접속정보를 기초로 해당 서비스 공급자에 접속하여 지도정보 및 위치, 센서, 구동체, POI와 실시간 센싱정보를 제공 받는다.

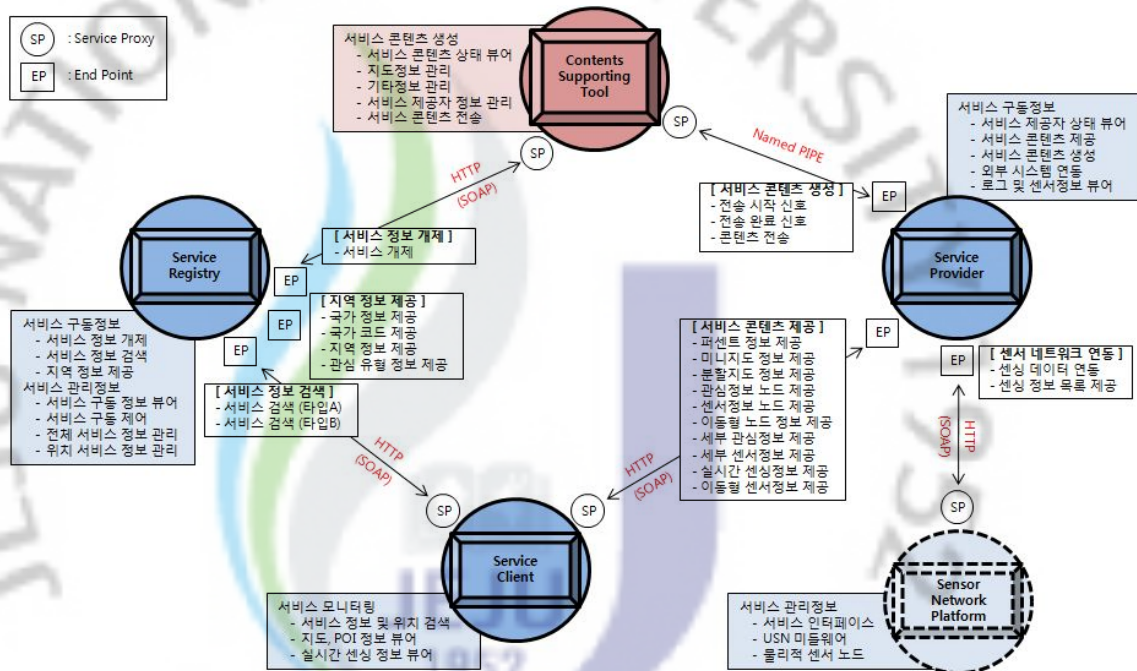


그림 12. 센서웹 OPEN API 정의 및 시스템 간 관계

그림 12는 센서웹의 OPEN API 정의를 나타낸다. OPEN API를 구현하기위해서 WCF기술을 적용하였으며, 서비스 제공은 End Point를 통해 서비스 요청은 Service Proxy를 이용한다. 서비스 저장소와 서비스 공급자는 COST, 서비스 요청자, 센서 네트워크 플랫폼을 위해 OPEN API를 제공하며, 각각의 OPEN API의 제공 정의는 다음과 같다.

서비스 저장소는 서비스 정보 등록, 지역 정보 제공, 서비스 정보 검색을 제공한다. 서비스 정보 등록은 COST에서 서비스 공급자의 정보를 등록하기위해 사용되며, 지역정보 제공은 COST와 서비스 요청자에서 지역정보(국가/지역/관심)를 제공 받을 수 있다. 서비스 정보 검색은 서비스 요청자에서 해당하는 서비스 공급자를 검색하기 위해서 사용한다. 서비스 공급자는 서비스 콘텐츠 생성과 서비스 콘텐츠 제공 및 센서 네트워크 연동을 제공한다. 서비스 콘텐츠 생성은 COST에서 콘텐츠(지도/위치매핑/POI/센서/구동체)를 생성하기 위해 사용한다. 서비스 콘텐츠 제공은 서비스 요청자에서 서비스 콘텐츠를 제공받기 위해 사용된다. 센서 네트워크 연동은 센서 네트워크 플랫폼의 실시간 센싱 정보를 제공받기 위해 사용한다.

## 2. 센서웹 구성요소

### 1) 서비스 저장소

#### (1) 서비스 저장소 주요기능

기능 분류	주요 기능	세부 구성 요소
외부 인터페이스	서비스 구동정보	서비스 정보 등록
		지역 정보 제공
		서비스 정보 검색
서비스 관리정보	서비스 구동정보 뷰어	
	서비스 구동 제어	서비스 정보 등록 제어
		지역 정보 제공 제어
		서비스 정보 검색 제어
	전체 서비스 정보 관리	등록 정보 관리
		미등록 정보 관리
		서비스 삭제 관리
	위치 서비스 정보 관리	국가정보 관리
		지역정보 관리
		관심유형 관리

표 2. 서비스 저장소의 주요기능

표 2는 서비스 저장소의 시스템 주요기능을 나타내고 있다. 서비스 저장소의 기능은 외부 인터페이스와 서비스 관리정보로 분류된다. 외부 인터페이스의 주요 기능은 서비스 구동정보가 있다. 서비스 관리정보의 주요 기능은 서비스 구동정보 뷰어, 서비스 구동 제어, 전체 서비스 정보 관리, 위치 서비스 정보 관리가 있다. 서비스 저장소에서 제공하는 서비스는 서비스 요청자와 COST를 위해 서비스 정보 등록, 지역 정보 제공, 서비스 정보 검색이 있다. 각각의 주요 기능은 그림 13에서 설명한다.

## (2) 서비스 저장소 설계

서비스 저장소를 구성하는 세부항목으로는 서비스 정보 등록, 서비스 정보 검색, 지역 정보, 서비스 구동정보 뷰어, 서비스 구동 제어, 전체 서비스 정보 관리, 위치 서비스 정보 관리, XML Configuration, 데이터베이스로 구성된다.

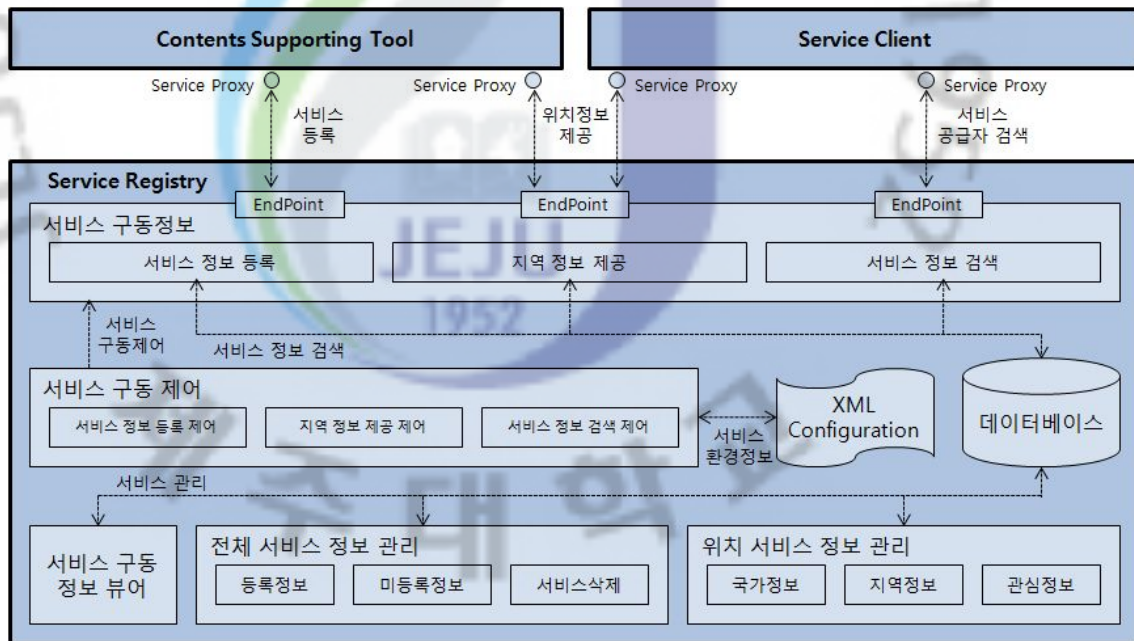


그림 13. 서비스 저장소 구성도

그림 13은 서비스 저장소 구성도를 나타낸다. 서비스 구동정보는 서비스 정보 등록, 지역 정보 제공, 서비스 정보 검색의 서비스를 제공하는 역할을 한다. 서비스 구동 제어는 XML Configuration에서 서비스 환경정보를 로드해 각각의 서비스를 제어하는 역할을 한다. 서비스 구동정보 뷰어는 서비스 구동정보의 서비스 정보 등록, 지역 정보 제공, 서비스 정보 검색의 구동여부 및 관리하는 서비스 정보를 모니터링 한다. 전체 서비스 정보 관리는 서비스 공급자의 등록정보 및 미등록정보를 관리하는 역할을 한다. 위치 서비스 정보 관리는 국가, 지역, 관심유형을 관리하는 기능을 한다. XML Configuration은 서비스 구동정보의 서비스 환경정보를 포함한다. 데이터베이스는 시스템 전반적인 데이터를 저장하는 역할을 한다.



그림 14는 서비스 저장소의 시퀀스 다이어그램이다. 서비스 저장소안의 내부 모듈이 서비스 요청자와 COST 연동과 내부 처리과정을 나타낸다.



그림 14. 서비스 저장소 시퀀스 다이어그램

서비스 저장소에서 서비스 구동정보 뷰어는 데이터베이스에 서비스 구동정보를 요청하고 응답을 받는다. XML 환경정보에서 서비스 구동정보(서비스 정보 등록, 지역 정보 제공, 서비스 정보 검색)의 접속정보를 로드하고 서비스 구동 제어에서 전체 서비스를 구동한다. COST는 서비스 저장소에게 지역 정보 및 서비스 정보등록에 대한 접속을 요청하고 응답한다. 이후 데이터베이스에서 요청 정보에 대해 검색 및 처리를 하고 처리결과를 반환한다. 서비스 요청자는 서비스 공급자에 대한 검색을 위해 서비스 저장소에 접속하고 서비스 구동제어 모듈은 검색 정보에 일치하는 항목을 데이터베이스에서 검색하고 결과를 반환한다. 전체 서비스 정보 관리에서는 전체 서비스 정보(서비스 명, 서비스 주소, 세부 항목, 검색 키워드, WSDL 제공여부, 등록여부, 국가정보, 지역정보, 관심유형)를 데이터베이스에 요청하고 응답을 받으며 해당 서비스 정보를 관리한다. 위치 서비스 정보 관리는 위치 정보(국가정보,

지역정보, 관심유형)를 데이터베이스에 요청하고 응답을 받으며 해당 서비스 정보를 관리한다.

그림 15는 서비스 저장소의 데이터베이스 객체 관계 다이어그램을 나타낸다.

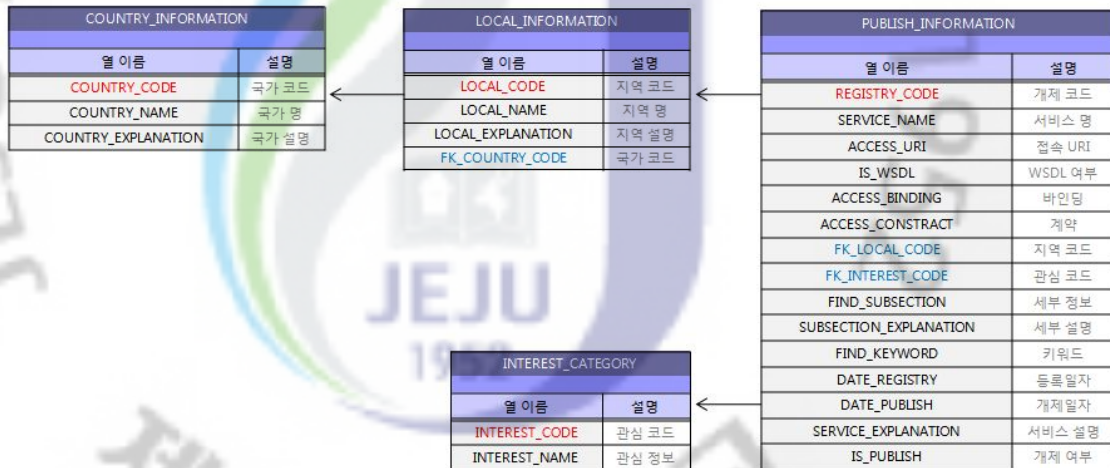


그림 15. 서비스 저장소 데이터베이스 객체 관계 다이어그램

데이터베이스의 각각의 테이블은 PUBLISH\_INFORMATION(등록정보), INTEREST\_CATEGORY(관심유형), LOCAL\_INFORMATION(지역정보), COUNTRY\_INFORMATION(국가정보)로 구성된다.

PUBLISH\_INFORMATION 테이블은 서비스 명, 서비스 설명, 접속 URI, WSDL 사용여부, 바인딩, 계약, 지역 및 POI, 세부정보 명, 세부정보 설명, 키워드, 등록일자, 개제일자, 개제여부 정보를 포함한다. INTEREST\_CATEGORY 테이블은 관심유형에 대한 정보를 포함한다. LOCAL\_INFORMATION 테이블은 지역 명, 지역 설명, 국가정보를 포함한다. COUNTRY\_INFORMATION 테이블은 국가 명, 국가 설명의 정보를 포함한다.

### (3) 서비스 저장소 구현

그림 16, 17은 서비스 저장소 구현의 클래스 다이어그램을 나타낸다. IServiceArea, IServiceFind, IServicePublish는 인터페이스는 나타내고, clsServiceArea, clsServiceFind, clsServicePublish는 인터페이스를 상속받는 서비스 구동정보 기능을 담당하는 클래스를 나타낸다. clsDatabaseControl은 데이터베이스를 제어하기 위한 클래스를 나타낸다. stAreaInformation, stServiceInformation는 프로그램에서 전역으로 사용하는 구조체를 나타낸다. frmMessageBox, frmAreaManagement, frmPublishManagement, frmServiceRegistry는 실제 화면에 표시되는 Windows Form을 상속받는 클래스를 나타낸다. frmMessageBox 클래스는 메시지를 나타내고, frmAreaManagement 클래스는 위치 서비스 정보 관리 기능을 담당한다. frmPublishManagement 클래스는 전체 서비스 정보 관리 기능을 담당하고, frmServiceRegistry 클래스는 서비스 구동정보 뷰어와 서비스 구동 제어 기능을 담당한다.

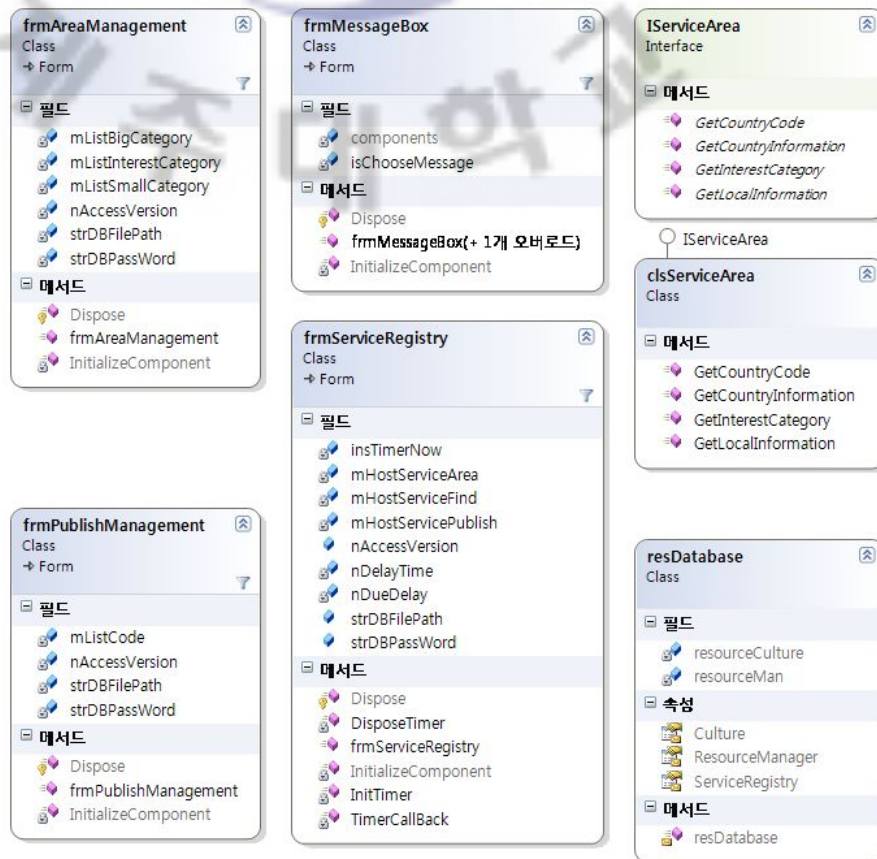


그림 16. 서비스 저장소의 클래스 다이어그램 1

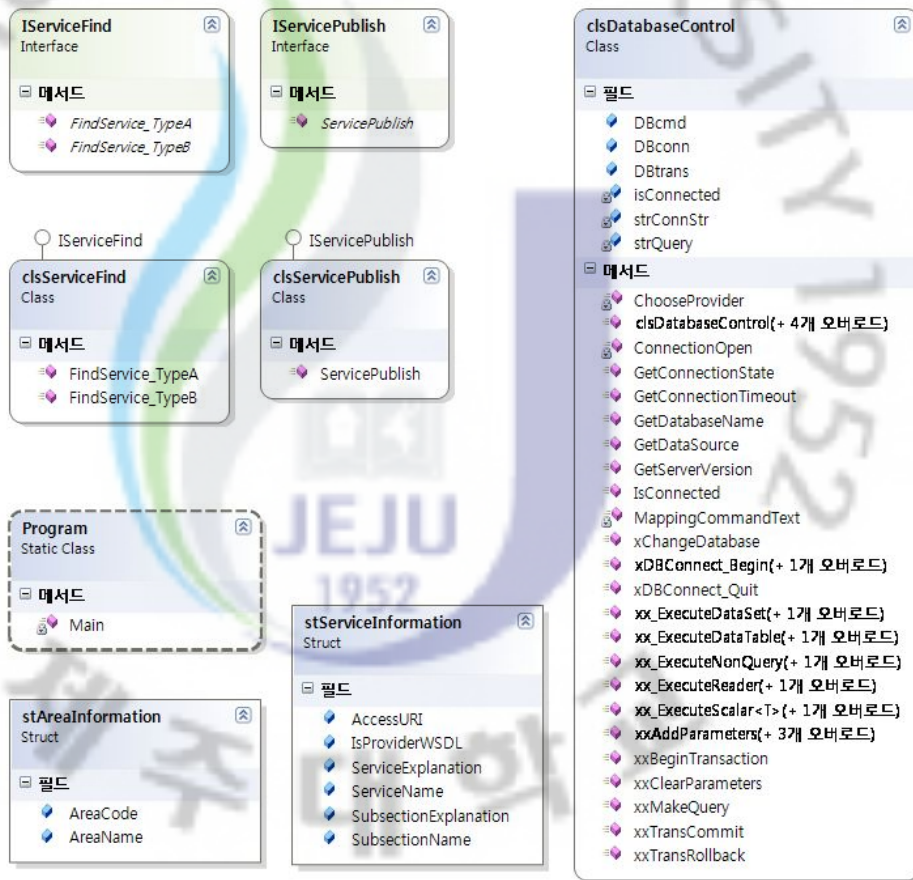


그림 17. 서비스 저장소의 클래스 다이어그램 2

서비스 저장소의 실행과정은 그림 18과 같다. 서비스 저장소는 서비스 등록소 메인화면, 위치정보/관심유형 관리화면, 서비스 정보 관리화면으로 나누어진다. 서비스 저장소 메인화면에서 위치 서비스 정보 관리를 선택하면 위치정보/관심유형 관리화면이 나타나고, 전체 서비스 정보 관리를 선택하면 서비스 정보 관리화면이 나타난다.

그림 19 좌측의 서비스 저장소 메인화면은 서비스 구동정보, 서비스 관리정보로 구성된다. 서비스 구동정보는 서비스의 상태 및 서비스 구동제어 할 수 있고 서비스 관리정보는 각각의 등록 된 서비스 정보를 보여주며 위치 서비스 정보관리 및 전체 서비스 정보를 관리할 수 있는 기능이 있다.

그림 19 우측의 위치정보/관심유형 관리화면은 국가정보, 위치정보 및 관심유형 관



리로 구성된다. 국가정보 관리는 서비스 공급자에 대한 국가의 정보를 관리하고 위치정보 관리는 각각의 국가에 대한 위치 정보를 관리한다. 관심유형은 사용자별로 교육기관, 행정기관 등에 대한 정보를 관리한다.

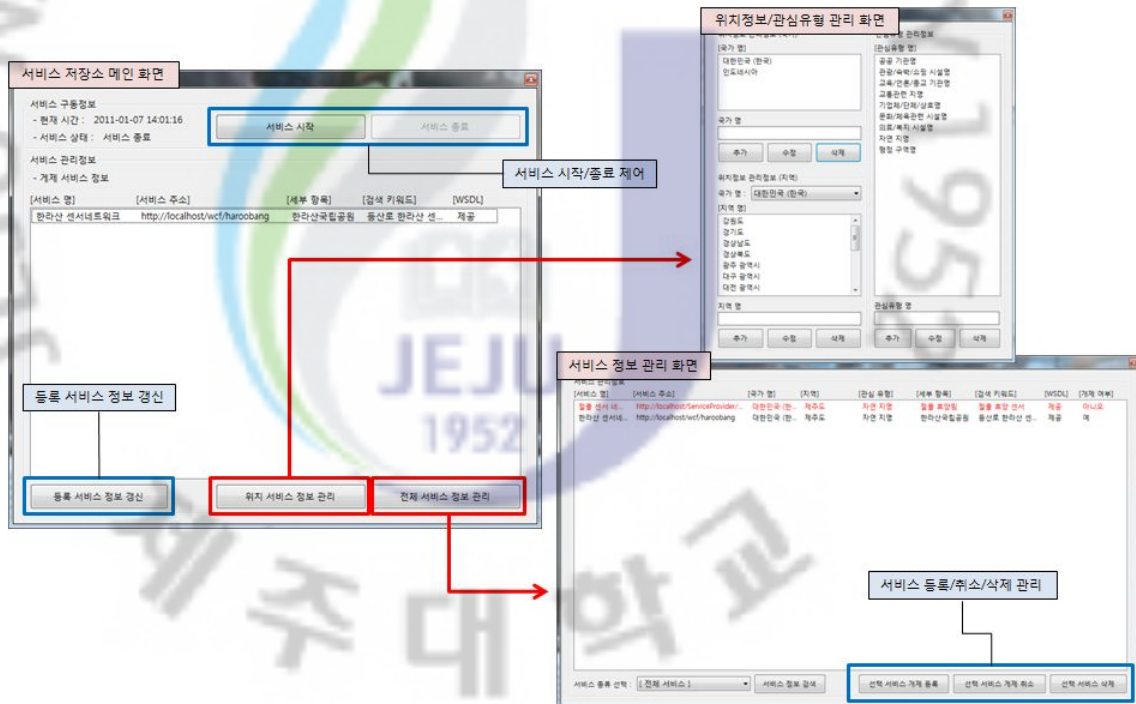


그림 18. 서비스 저장소의 실행과정

그림 20의 서비스 정보 관리화면은 서비스 관리정보 뷰어, 서비스 종류 검색, 등록 관리로 구성된다. 서비스 관리정보 뷰어는 서비스 공급자 검색을 위한 공급자의 등록정보(서비스명, 서비스 주소, 국가명, 지역, 관심유형, 세부항목, 검색 키워드, WSDL 여부, 등록 여부)를 관리한다. 서비스 종류 검색은 종류별 서비스를 선택하여 해당 서비스 정보를 검색한다. 등록 관리는 선택 서비스의 등록을 확인/취소하고 삭제할 수 있는 기능을 제공한다.

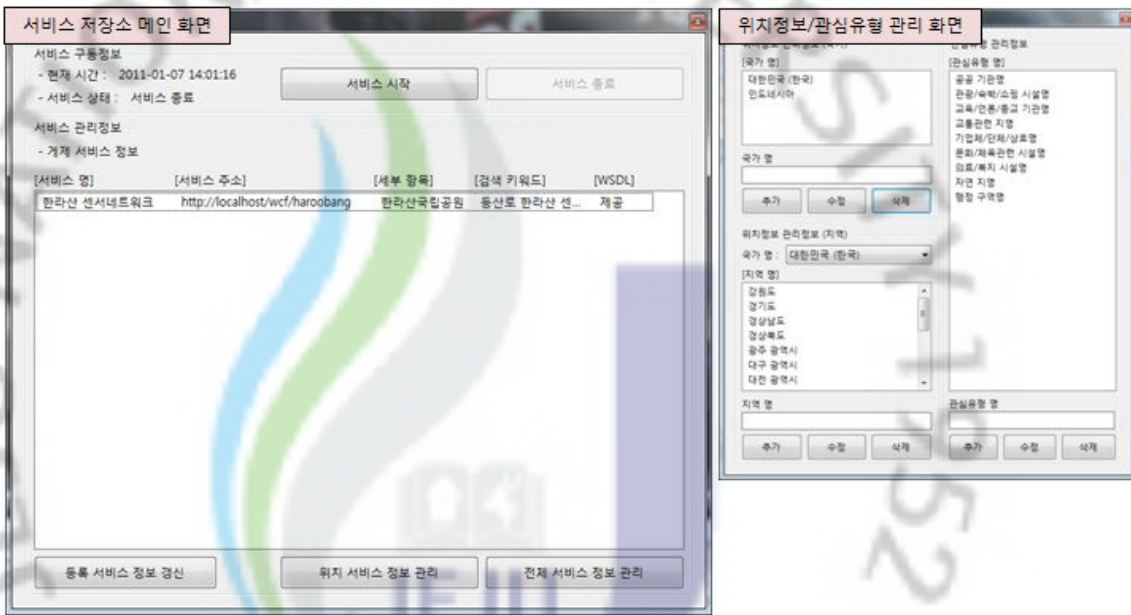


그림 19. 서비스 등록 및 위치 서비스 정보 관리 실행화면

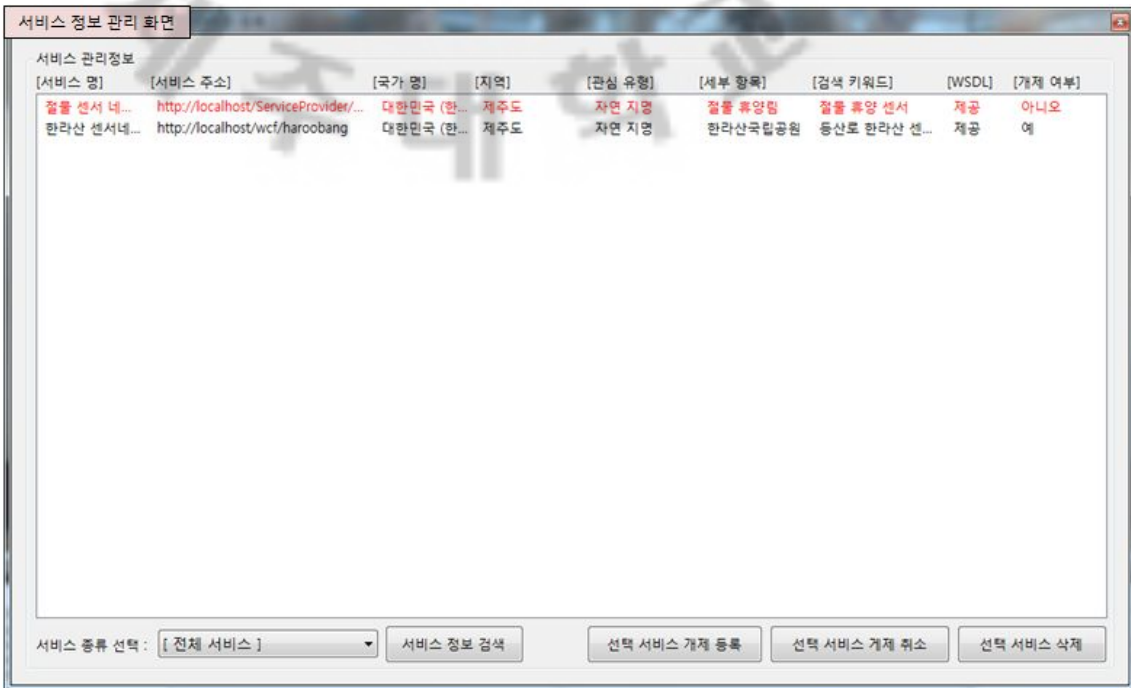


그림 20. 전체 서비스 정보 관리 실행화면

## 2) 서비스 요청자

(1) 서비스 요청자 주요기능

기능 분류	주요 기능	세부 구성 요소
서비스 모니터링	서비스 위치 정보	국가 정보 요청
		지역 정보 요청
		관심 분류 요청
	서비스 검색	서비스 URI 검색
	서비스 콘텐츠 요청	지도 정보 요청
		노드 정보 요청
		실시간 센싱 정보 요청
	서비스 검색 결과 뷰어	
	뷰어 제어	지도 이동
		지도 확대/축소
		노드 숨기기/보이기
	서비스 모니터링 뷰어	지도 정보 뷰어
		POI 노드 뷰어
		센서 노드 뷰어
		구동체 노드 뷰어
노드 정보 뷰어		
실시간 센싱정보 뷰어		

표 3. 서비스 요청자의 주요기능

표 3은 서비스 요청자의 시스템 주요기능을 나타낸다. 서비스 요청자의 기능은 서비스 모니터링으로 분류된다. 서비스 모니터링의 주요기능은 서비스 위치 정보, 서비스 검색, 서비스 콘텐츠 요청, 서비스 검색 결과 뷰어, 뷰어 제어, 서비스 모니터링 뷰어가 있다. 서비스 요청자에서 서비스 위치 정보 및 검색은 서비스 저장소의 OPEN API를 이용하여 위치정보와 서비스 정보를 검색하고 서비스 검색결과를 표현한다. 서비스 모니터링은 서비스 검색의 서비스 검색결과를 기초로 서비스 공급자에서의 OPEN API를 이용하여 지도 정보 및 노드(POI, 센서, 구동체)정보 및 각각의 노드에 대한 상세정보(노드명 노드설명, 좌표, 이미지, UFID, 센싱목록), 센서 네트워크 플랫폼의 실시간 센서정보(온도, 습도, 조도, 위치 전력량)를 제공 받고 정보를 화면에 표현한다. 각각의 주요 기능은 그림 21에서 설명한다.

## (2) 서비스 요청자 설계

서비스 요청자를 구성하는 세부항목으로는 서비스 위치 정보, 서비스 검색, 서비스 콘텐츠 요청, 뷰어 제어, 서비스 검색 결과 뷰어, 서비스 모니터링 뷰어, 콘텐츠 정보 메모리 테이블로 구성된다.

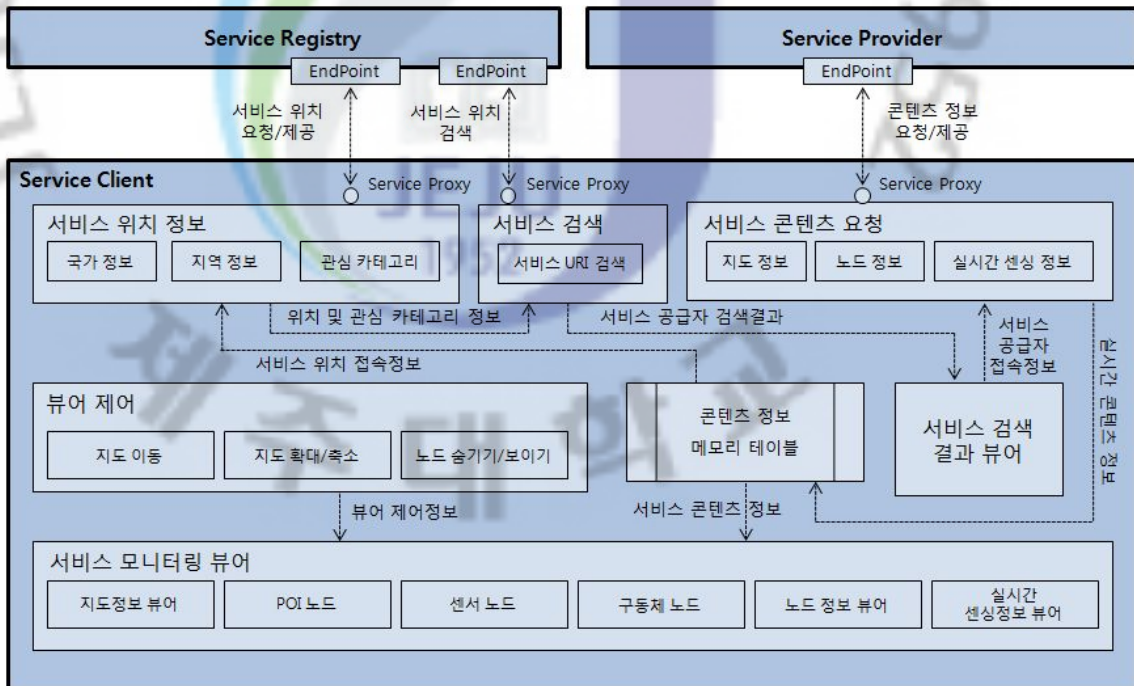


그림 21. 서비스 요청자 구성도

그림 21은 서비스 요청자 구성도를 나타낸다. 서비스 위치정보는 콘텐츠 정보 메모리 테이블에서 서비스 위치 접속정보를 로드하고 국가정보, 지역정보, 관심 분류 정보를 제공 받는다. 서비스 검색은 위치 및 관심 분류 정보를 기초로 서비스 URI를 검색한다. 서비스 검색 결과 뷰어에서 검색 결과를 표시하고 해당 서비스로 접속한다. 서비스 콘텐츠 요청은 지도정보, 노드정보, 실시간 센싱 정보를 제공받는다. 서비스 모니터링 뷰어는 지도정보 뷰어, POI노드, 센서노드, 구동체노드, 노드정보 뷰어, 실시간 센싱정보 뷰어 기능이 있다. 뷰어 제어는 지도의 이동, 확대, 축소 및 노드 제어 기능이 있다. 콘텐츠 정보 메모리 테이블은 실시간 콘텐츠 정보를 저장

하는 역할을 한다.

그림 22는 서비스 요청자의 시퀀스 다이어그램 이다. 서비스 요청자안의 내부 모듈이 서비스 저장소와 서비스 공급자와의 연동과 내부 처리과정을 나타낸다.



그림 22. 서비스 요청자 시퀀스 다이어그램

서비스 검색에서 서비스 저장소에게 국가정보, 지역정보, 관심 분류정보를 요청하고 응답 받는다. 공급자의 서비스 목록을 요청하고 콘텐츠 정보 메모리 테이블에 그 결과를 저장하며 지도뷰어로 서비스 URI를 전송한다. 지도뷰어에서는 서비스 공급자에게 지도 정보, 노드좌표(POI, 센서, 구동체)를 요청하고 정보를 제공받으며 각각의 뷰어에서는 콘텐츠 정보 메모리 테이블안의 정보를 토대로 지도 및 노드정보를 그려준다. 노드정보 뷰어에서는 서비스 공급자에게 선택 된 노드에 대한 상세 정보를 요청하고 응답받은 내용을 출력한다. 노드가 선택이 되고 실시간 센서 뷰어에서는 서비스 공급자에게 실시간 센싱정보 목록을 제공받으며 이를 출력한다.



### (3) 서비스 요청자 구현

그림 23, 24는 서비스 요청자 구현의 클래스 다이어그램을 나타낸다. IServiceFind, IServiceArea는 인터페이스를 나타내고, ServiceFindClient, ServiceAreaClient는 인터페이스를 상속받는 서비스 클래스를 나타낸다. IServiceProvider는 인터페이스를 나타내고, ServiceProviderClient는 인터페이스를 상속받는 서비스 구동의 클래스를 나타낸다. StAreaInformation, stDivisionMap, sSensingInformation, stNodeDetail\_Sensor, stNodeInformation, stMiniMapInformation, stPercentInformation, stNodeDetail\_PointOfInterest는 프로그램에서 전역으로 사용되는 구조체를 나타내고, eKindofNode는 전역으로 사용되는 열거형 상수를 나타낸다. SearchMain, ViewerMain은 서비스 검색 부분의 화면을 구성하고 Windows Form을 상속받는 클래스를 나타낸다. SearchMain 클래스는 서비스 위치정보, 서비스 검색, 서비스 검색 결과 뷰어의 기능을 담당한다. ViewerMain 클래스는 서비스 콘텐츠 요청, 뷰어 제어, 서비스 모니터링 뷰어의 기능을 담당한다.

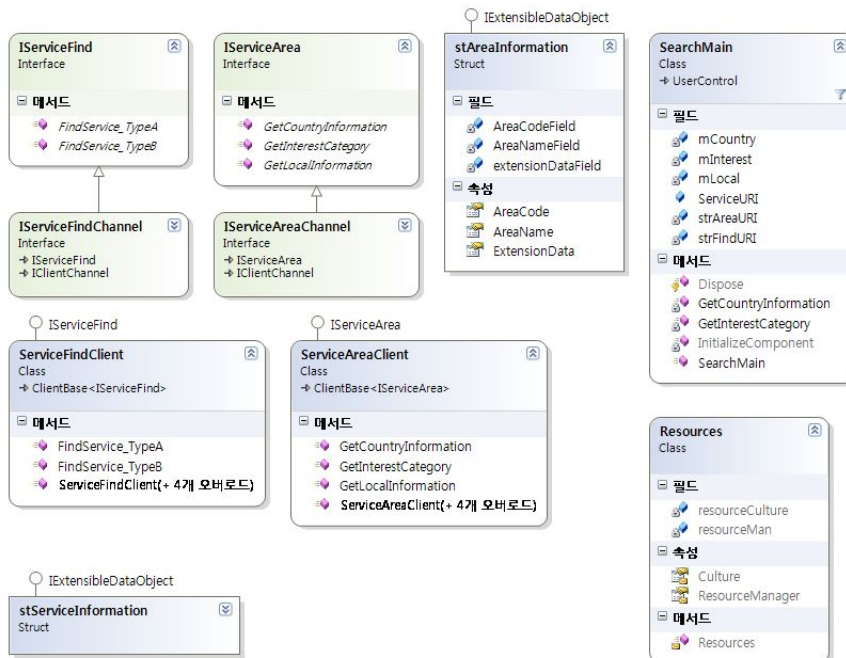


그림 23. 서비스 요청자의 클래스 다이어그램 1

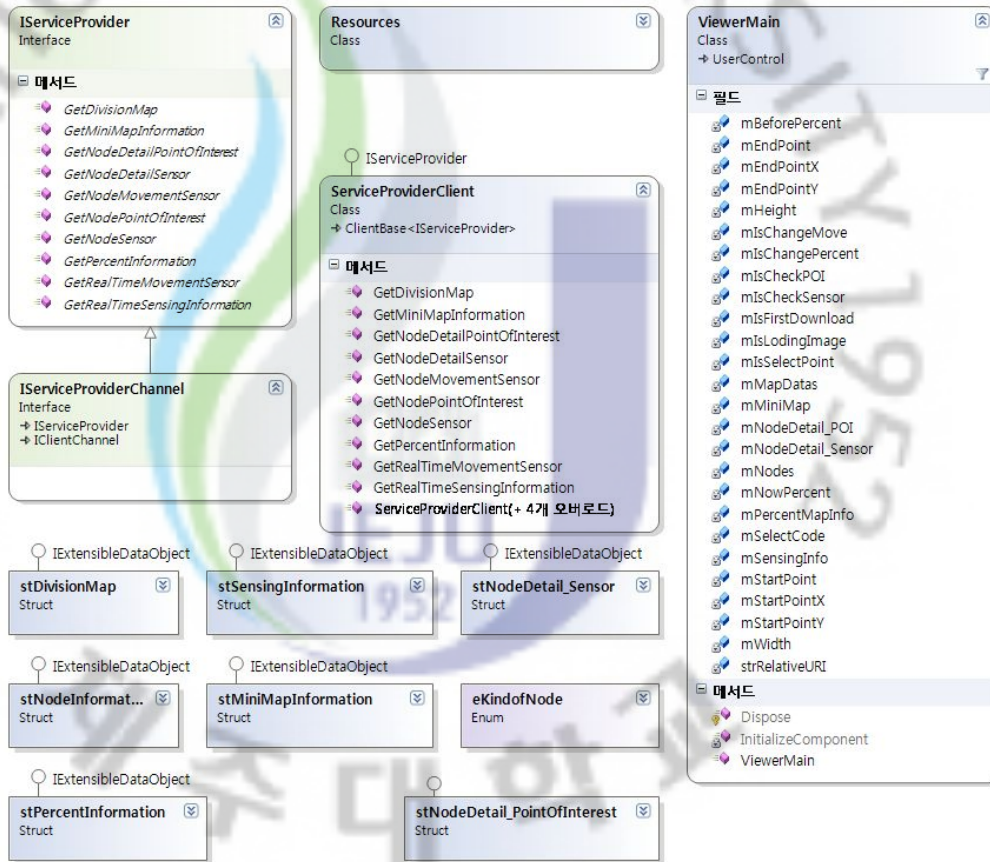


그림 24. 서비스 요청자의 클래스 다이어그램 2

서비스 요청자의 실행과정은 그림 25와 같다. 서비스 요청자의 실행은 기본적으로 인터넷 브라우저에서 동작하며, Smart Client를 통해 배포된다. 서비스 요청자는 크게 서비스 검색화면과 콘텐츠 뷰어화면으로 구분된다. 콘텐츠 뷰어 화면에서 지도 검색을 선택하면 서비스 검색 화면이 나타난다.

그림 26의 서비스 검색화면은 공급자를 검색하는 화면이고 서비스 저장소와 서비스 검색의 기능으로 나누어진다. 서비스 저장소는 지역정보를 서비스하는 URI와 검색 서비스의 URI를 설정할 수 있다. 서비스 검색은 국가명, 지역명, 관심 분류, 검색 세부항목, 검색 키워드로 해당하는 공급자 정보를 검색하고 결과 목록을 나타낸다. 검색된 서비스 목록을 클릭하여 해당 공급자에 접속하여 서비스를 제공받는다.

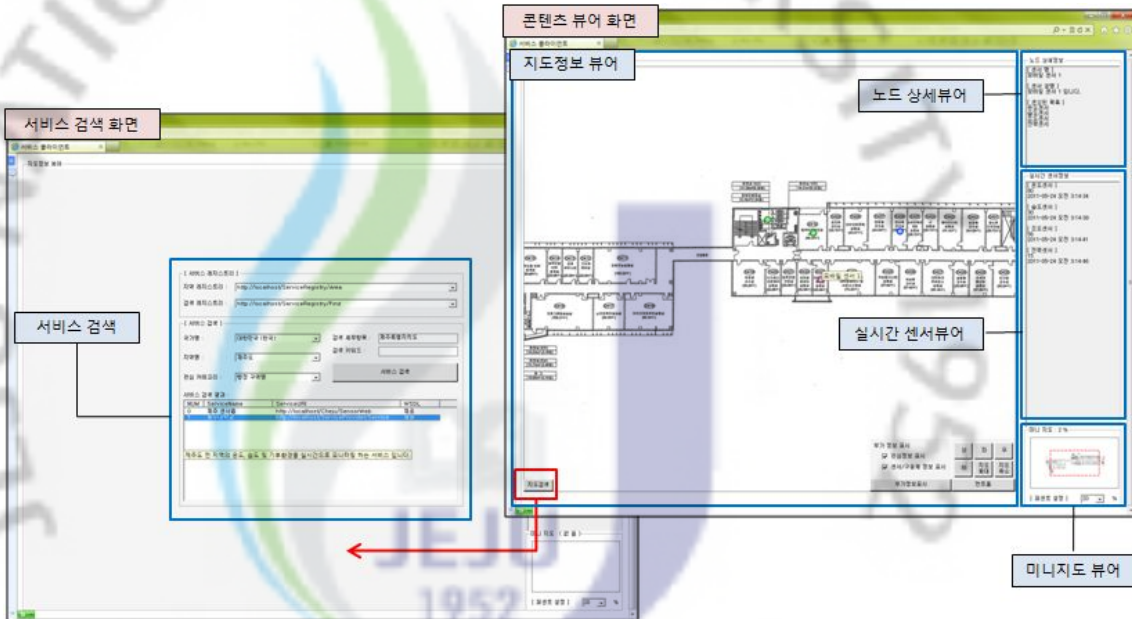


그림 25. 서비스 요청자의 실행과정

그림 27의 콘텐츠 뷰어화면은 서비스 공급자에 접속하여 콘텐츠를 제공받고 표현하는 역할을 한다. 콘텐츠 뷰어화면은 지도정보 뷰어, 노드 상세뷰어, 실시간 센서뷰어, 미니지도 뷰어로 나뉘어진다. 지도정보 뷰어는 지도정보 및 노드정보 제어부와 서비스 검색기능, 지도 및 노드정보를 표현하는 영역이 있다. 노드 상세뷰어는 지도정보 뷰어에서 노드를 클릭하고 표현되는 영역이고 표현되는 정보는 노드명, 노드 설명, 센싱 목록 및 세부 노드정보가 있다. 실시간 센서 뷰어는 선택된 노드의 실시간 센싱정보(온도, 습도, 조도, 위치, 전력량 등)를 표현한다. 미니지도 뷰어는 해당 퍼센트의 지도에서 지도정보 뷰어에서 표현되는 영역을 보여주고 현재 지도의 퍼센트를 설정할 수 있다.

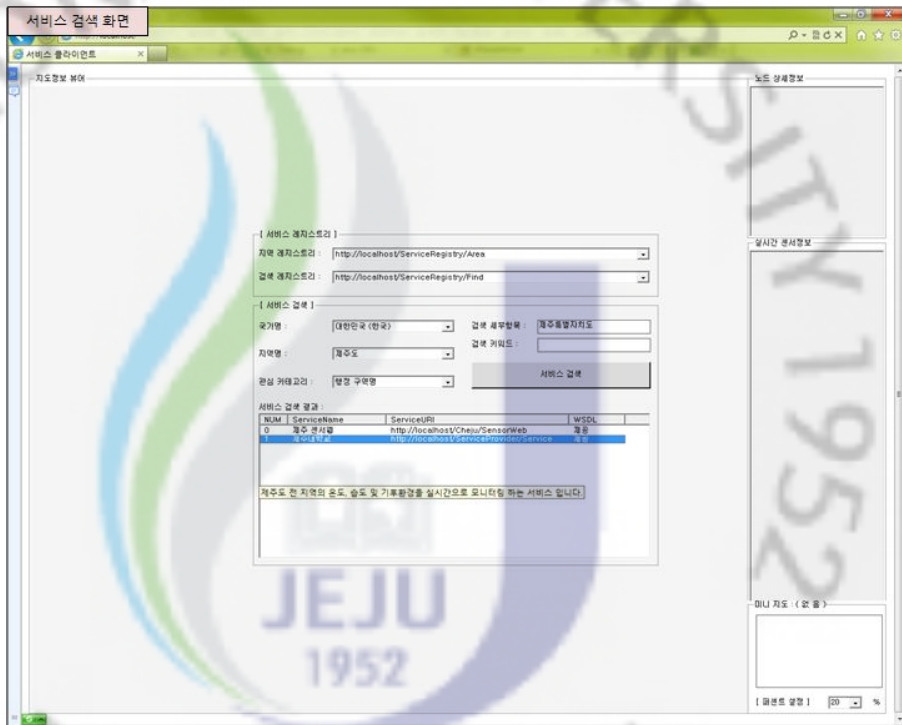


그림 26. 서비스 검색 실행화면

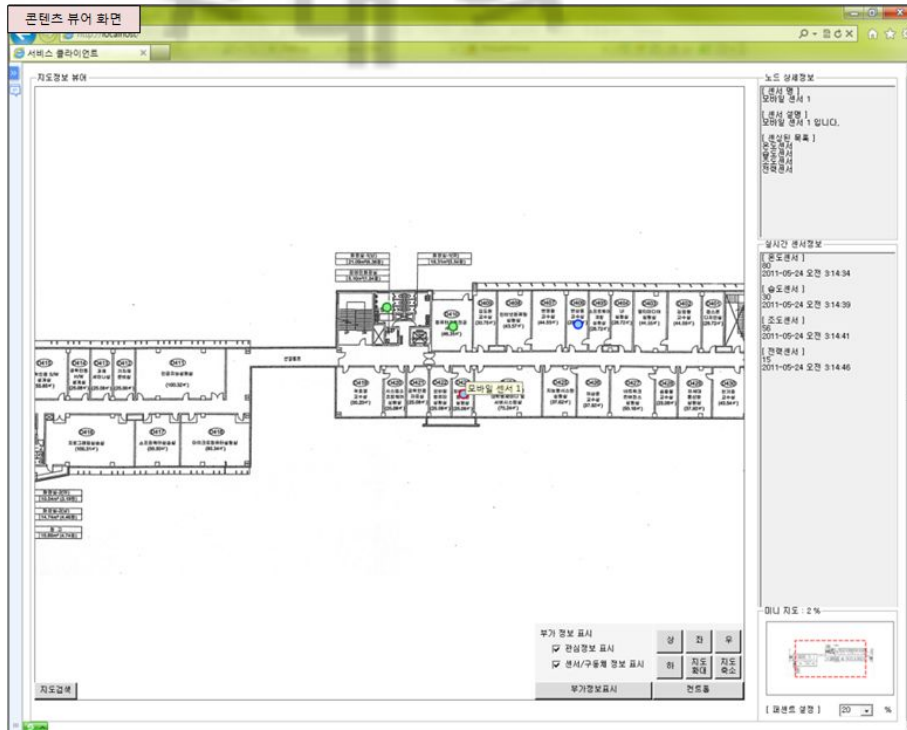


그림 27. 콘텐츠 뷰어 실행화면



### 3) 서비스 공급자

#### (1) 서비스 공급자 주요기능

기능 분류	주요 기능	세부 구성 요소
외부 인터페이스	서비스 구동정보	서비스 콘텐츠 제공
		서비스 콘텐츠 생성
		센서 네트워크 연동
서비스 관리정보	서비스 구동 제어	서비스 콘텐츠 제공 제어
		서비스 콘텐츠 생성 제어
		센서 네트워크 연동 제어
	서비스 공급자 상태 뷰어	
	서비스 로그 뷰어	
	센서정보 메모리 뷰어	
프로그램 배포관리	서비스 공급자를 위한 운영 프로그램 수신부	

표 4. 서비스 공급자의 주요기능

표 4는 서비스 공급자의 시스템 주요기능을 나타낸다. 서비스 공급자의 기능은 외부 인터페이스, 서비스 관리정보, 프로그램 배포관리로 분류된다. 외부 인터페이스의 주요 기능은 서비스 구동정보이다. 서비스 관리정보와 주요 기능은 서비스 구동 제어, 서비스 공급자 상태 뷰어, 서비스 로그 뷰어, 센서정보 메모리 뷰어가 있다. 프로그램 배포관리의 주요 기능은 서비스 공급자를 위한 운영 프로그램 수신부가 있다. 서비스 공급자에서 제공하는 서비스는 서비스 요청자와 COST, 센서 네트워크 플랫폼을 위해 서비스 콘텐츠 제공, 서비스 콘텐츠 생성, 센서 네트워크 연동이 있다. 각각의 주요 기능은 그림 28에서 설명한다.



## (2) 서비스 공급자 설계

서비스 공급자는 운영 프로그램 수신부를 통해 해당 프로그램을 다운로드 받고 구동한다. 서비스 공급자를 구성하는 세부항목으로는 서비스 구동정보, 서비스 구동 제어, XML Configuration, 데이터베이스, 센서정보 메모리 테이블, 서비스 공급자 상태 뷰어, 서비스 로그 뷰어, 센서정보 메모리 뷰어로 구성된다.

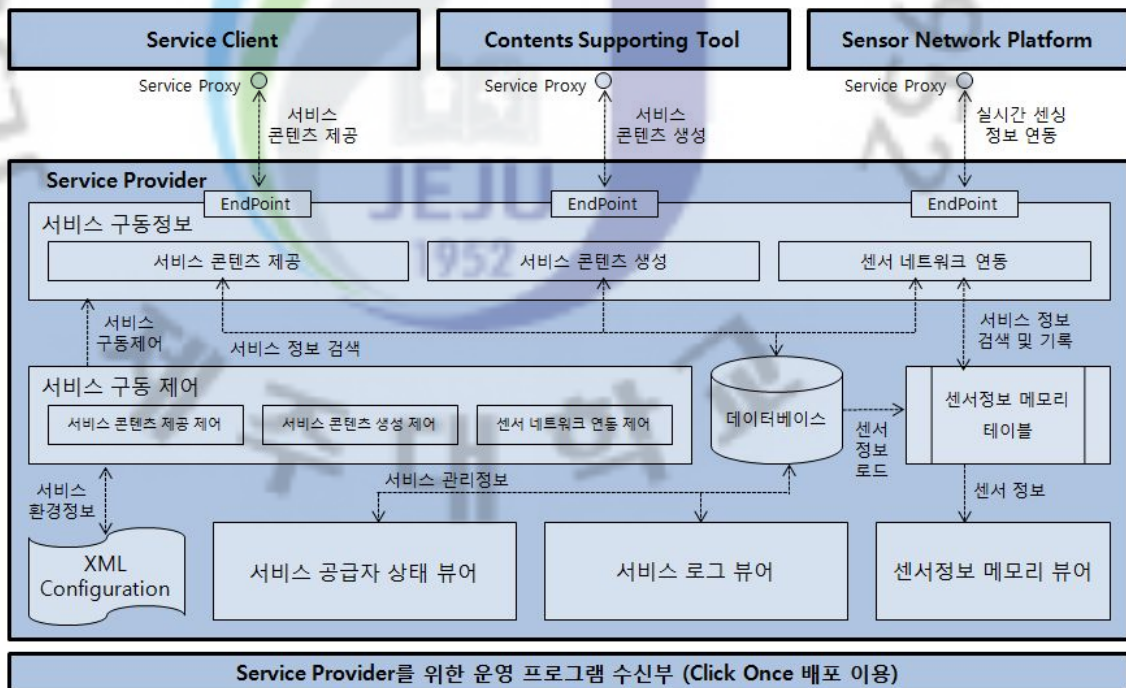


그림 28. 서비스 공급자 구성도

그림 28은 서비스 공급자 구성도를 나타낸다. 서비스 공급자는 웹 브라우저를 이용한 운영 프로그램 수신부로 COST로부터 프로그램을 다운로드 받고 구동한다. 서비스 구동정보는 서비스 콘텐츠 제공, 서비스 콘텐츠 생성, 센서 네트워크 연동의 서비스를 제공하는 역할을 한다. 서비스 구동 제어는 XML Configuration에서 서비스 환경정보를 로드해 각각의 서비스를 제어하는 역할을 한다. 서비스 공급자 상태 뷰어는 서비스 공급자의 상태를 모니터링한다. 서비스 로드 뷰어는 서비스의 전반적인 구동상태의 로그를 관리하는 역할을 한다. 센서정보 메모리 뷰어는 센서정보

메모리 테이블의 상태를 모니터링하는 역할을 한다. XML Configuration은 서비스 구동정보의 서비스 환경정보를 포함한다. 데이터베이스는 시스템 전반적인 데이터를 저장하는 역할을 한다. 센서정보 메모리 테이블은 데이터베이스에서 센서정보를 로드하여 메모리 테이블을 구성하고 이를 기초로 센서 네트워크 플랫폼에서 들어오는 실시간 센싱 정보를 저장하는 역할을 한다.

그림 29는 서비스 공급자의 시퀀스 다이어그램이다. 서비스 공급자안의 내부 모듈이 COST와 서비스 요청자, 센서 네트워크 플랫폼과 내부 처리과정을 나타낸다.

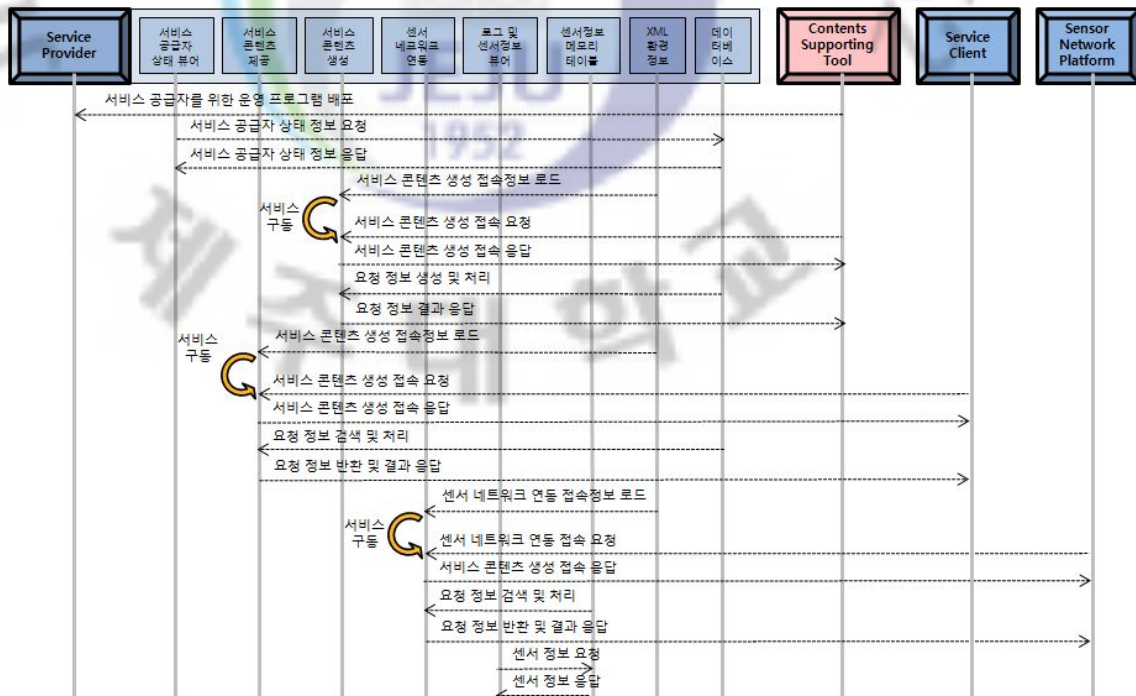


그림 29. 서비스 공급자 시퀀스 다이어그램

서비스 공급자는 COST를 통해 운영 프로그램을 배포받고 구동한다. 서비스 공급자에서 서비스 공급자 상태 뷰어는 데이터베이스에 공급자 상태정보를 요청하고 응답을 받는다. 서비스 콘텐츠 생성을 위해 XML 환경정보에서 서비스 구동정보를 로드하고 서비스 콘텐츠 생성에서 서비스를 구동한다. COST는 콘텐츠 생성 및 요청을 수행하고 서비스 콘텐츠 생성 모듈에서 처리 결과를 응답한다. 서비스 요청자에

계 서비스 콘텐츠 정보를 제공하기 위해 서비스 콘텐츠 제공은 XML 환경정보를 로드하고 서비스를 구동한다. 서비스 요청자는 콘텐츠를 제공받기 위해 서비스 공급자에게 접속요청을 하고 응답을 받는다. 센서 네트워크 연동 모듈은 센서 네트워크 플랫폼과 연동하기 위해 XML 환경정보를 로드하고 서비스를 구동한다. 센서 네트워크 플랫폼은 서비스 공급자에게 접속 요청하고 응답받는다. 또한 센서정보 메모리 테이블에서는 센서 네트워크 플랫폼의 요청 정보에 대한 검색 및 처리를 한다.

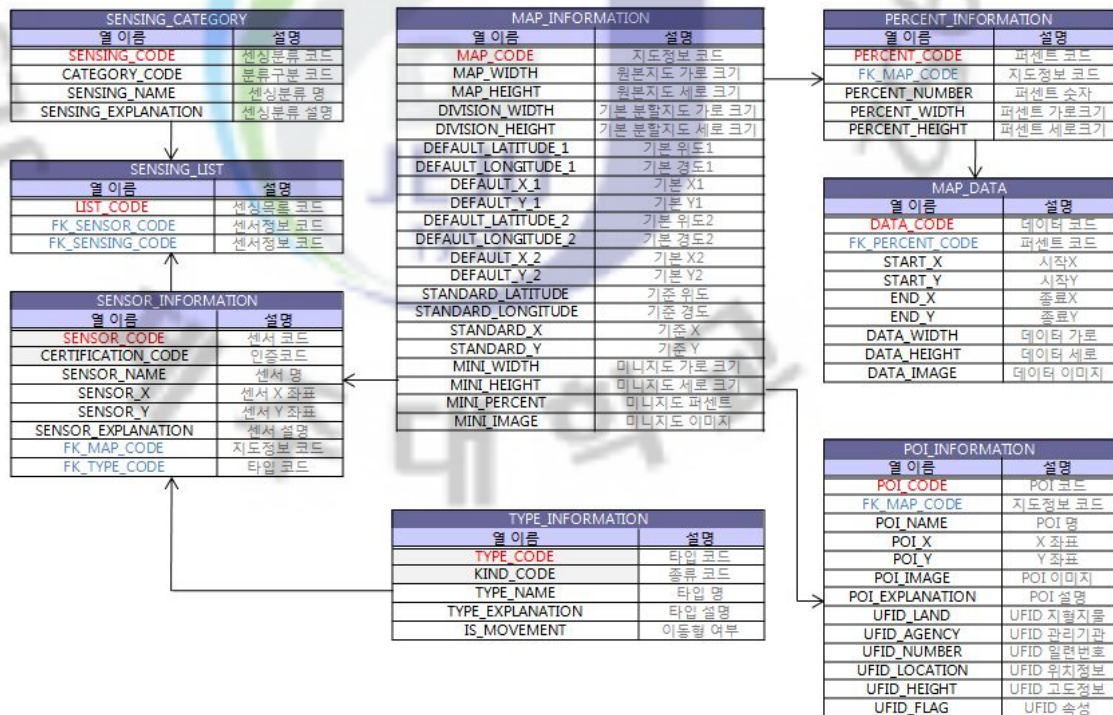


그림 30. 서비스 공급자 데이터베이스 객체 관계 다이어그램

그림 30은 서비스 공급자의 데이터베이스 객체 관계 다이어그램을 나타낸다. 데이터베이스의 각각의 테이블은 MAP\_INFORMATION(지도정보), SENSING\_CATEGORY(센싱분류), SENSING\_LIST(센싱목록), SENSOR\_INFORMATION(센서정보), TYPE\_INFORMATION(타입설명), PERCENT\_INFORMATION(퍼센트정보), MAP\_DATA(분할지도 데이터), POI\_INFORMATION(관심정보)로 구성된다. MAP\_INFORMATION 테이블은 원본지도 데이터정보, 위치매핑 정보, 미니지도 정보를 포함한다. SENSING\_CATEGORY 테이블은 센싱분류 정보를 하고 SENSOR\_INFORM

ATION 테이블은 센서의 픽셀좌표 및 센서정보를 포함한다. TYPE\_INFORMATION 테이블은 각각의 센서에 대한 정보를 포함한다. SENSING\_LIST 테이블은 동적으로 하나의 센서에 대한 동적인 센싱분류 목록을 작성하기 위한 정보를 포함한다. PERCENT\_INFORMATION 테이블은 각각의 분할지도 정보에 대한 퍼센트 정보를 포함한다. MAP\_DATA 테이블은 분할 된 지도정보를 포함한다. POI\_INFORMATION 테이블은 POI 노드에 대한 픽셀좌표와 UFID정보를 포함한다.

그림 31은 서비스 공급자가 이동형 센서 및 구동체 노드를 위해 센서 네트워크 플랫폼에서 실시간으로 들어오는 위치정보를 픽셀좌표로 매핑하는 과정을 나타낸다.

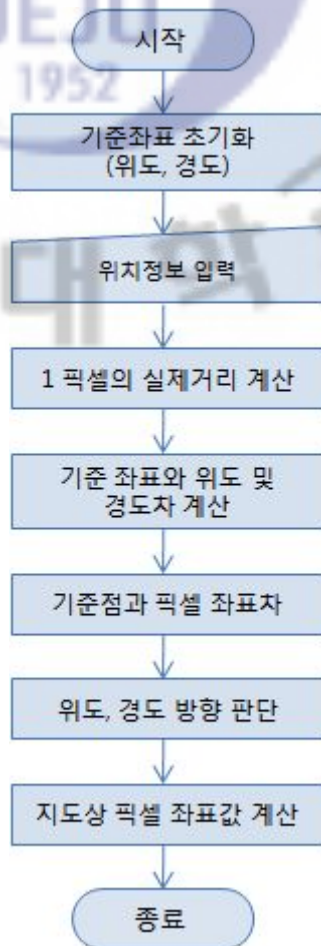


그림 31. 위도/경도와 화면 픽셀좌표의 매핑 순서도



기본적으로 매핑을 위해서는 기준좌표가 초기화 되어야 한다. 기준좌표는 위치정보(위도, 경도)와 픽셀좌표가 매핑 된 좌표 이며 실제 계산에서 기준값 역할을 한다. 위치정보 매핑은 해당노드의 픽셀좌표를 나타내기 위해 센서 네트워크 플랫폼에서 실시간으로 들어오는 위치정보를 계산하는 과정을 거친다. 위치정보 매핑을 위해 센서 네트워크 플랫폼에서 전송 된 위치정보를 통해 1픽셀의 실제거리를 계산하고 기준좌표를 통해 위치 및 경도차를 계산하고 기준점과 픽셀 좌표차를 계산한다. 위도, 경도의 방향 판단을 통해 지도상의 픽셀 좌표값을 계산한다.

그림 32는 위치정보를 픽셀좌표와 매핑하기 위한 매핑 시작점, 끝점, 기준점과 해당 노드의 정보를 설명한다. 그림 32의 매핑 순서도에서 실제 매핑을 수행하고 연산하기 위해서 쓰이는 A(매핑 시작점), B(매핑 끝점), C(매핑 기준점), D(해당 노드)를 설명한다. 매핑 시작점의 픽셀좌표는 APixX(X좌표), APixY(Y좌표), 위치정보는 ALon(경도), ALat(위도)로 나타낸다. 매핑 끝점의 픽셀좌표는 BPixX(X좌표), BPixY(Y좌표), 위치정보는 BLon(경도), BLat(위도)로 나타낸다. 매핑 기준점의 픽셀좌표는 CPixX(X좌표), CPixY(Y좌표), 위치정보는 CLon(경도), CLat(위도)로 나타낸다. 해당 노드의 픽셀좌표는 DPixX(X좌표), DPixY(Y좌표), 위치정보는 DLon(경도), DLat(위도)로 나타낸다. 매핑 시작점과 끝점, 기준점은 이미 값이 설정 및 초기화 되어있고, 해당 노드의 픽셀정보를 나타내기 위해 입력으로 들어온 위치정보(DLon, DLat)을 매핑 기준점과 연산해 픽셀좌표(DPixX, DPixY)를 출력한다.

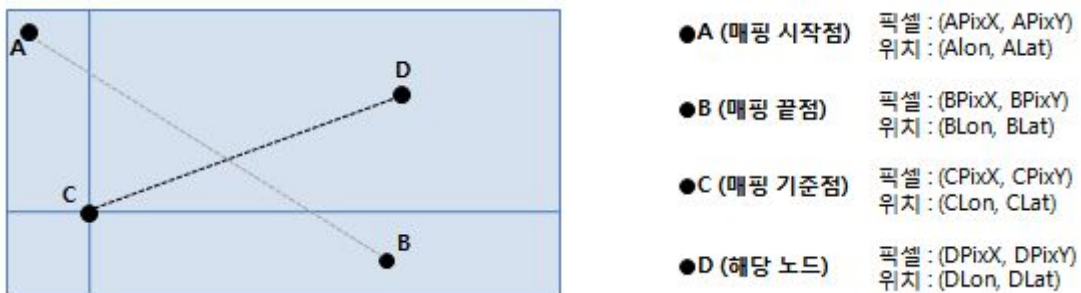


그림 32. 위치정보 매핑 설명



수식 1은 1 픽셀의 실제거리 계산에서 사용되는 수식이다. PixDistanceX는 AB간 X방향의 픽셀 거리차이고, PixDistanceY는 AB간 Y방향의 픽셀 거리차이다. PixDistance는 AB간 픽셀좌표의 거리를 뜻하고, Distance는 AB간 실제거리를 나타낸다. Radio는 1픽셀의 실제거리를 나타낸다.

$$\begin{aligned}
 \text{PixDistanceX} &= |A\text{PixX} - B\text{PixX}| \\
 \text{PixDistanceY} &= |A\text{PixY} - B\text{PixY}| \\
 \text{PixDistance} &= \sqrt{\text{PixDistanceX}^2 + \text{PixDistanceY}^2} \\
 \text{Distance} &= 69.1 \times (180/\text{PI}) \times \arccos[\sin(\text{ALat}) \times \sin(\text{BLat}) + \cos(\text{ALat}) \times \cos(\text{BLat}) \times \cos(\text{BLon} - \text{ALon})] \\
 \text{Radio} &= \text{Distance} / \text{PixDistance}
 \end{aligned}
 \tag{수식 1}$$

수식 2는 기준 좌표와 위도 및 경도차 계산에서 사용되는 수식이다. DistanceLat은 CD간 위도방향의 거리차를 뜻하고, DistanceLon은 CD간 경도방향의 거리차를 나타낸다.

$$\begin{aligned}
 \text{DistanceLat} &= 69.1 \times |D\text{Lat} - C\text{Lat}| \\
 \text{DistanceLon} &= 69.1 \times |D\text{Lon} - C\text{Lon}| \times \cos(D\text{Lat} / 57.3)
 \end{aligned}
 \tag{수식 2}$$

수식 3은 기준점과 픽셀 좌표차 계산에서 사용되는 수식이다. PixDistanceX\_CD는 CD간 X방향의 픽셀 거리차를 뜻하고, PixDistanceY\_CD은 CD간 Y방향의 픽셀 거리차를 나타낸다.

$$\begin{aligned}
 \text{PixDistanceX\_CD} &= \text{DistanceLon} / \text{Radio} \\
 \text{PixDistanceY\_CD} &= \text{DistanceLat} / \text{Radio}
 \end{aligned}
 \tag{수식 3}$$

수식 4는 지도상 픽셀 좌표값 계산에서 사용되는 수식이다. DPixX는 해당 노드의 픽셀 X좌표를 뜻하고, DPixY는 해당 노드의 Y 좌표를 뜻한다. 또한 위도, 경도의 방향 판단을 통해 기준점과 픽셀 좌표차의 연산이 선택된다.

$$\begin{aligned}
 \text{DPixX} &= \text{CPixX} \pm \text{PixDistanceX\_CD} \\
 \text{DPixY} &= \text{CPixY} \pm \text{PixDistanceY\_CD}
 \end{aligned}
 \tag{수식 4}$$

### (3) 서비스 공급자 구현

그림 33, 34은 서비스 공급자 구현의 클래스 다이어그램을 나타낸다.

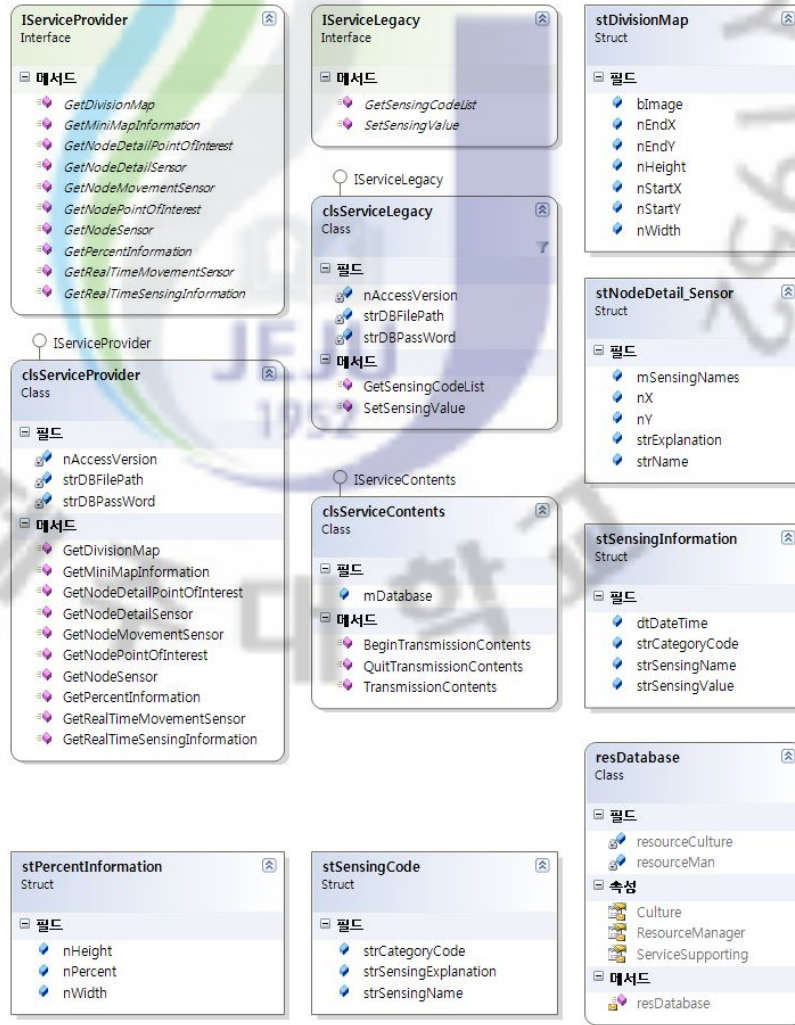


그림 33. 서비스 공급자의 클래스 다이어그램 1

IServiceProvider, IServiceLegacy는 인터페이스를 나타내고, clsServiceProvider, clsServiceLegacy, clsServiceContents는 인터페이스를 상속받는 서비스 구동의 클래스를 나타낸다. clsDatabaseControl은 데이터베이스를 제어하기 위한 클래스를 나타낸다. stPercentInformation, stSensingCode, stDivisionMap, stNodeDetail\_Sensor, stS

ensingInformation, stMiniMapInformation, stNodeInformation, stNodeDetail\_PointOfInterest, eKindofNode는 전역으로 사용하는 구조체 및 열거형을 나타낸다. frmSensorMemoryView, frmServiceProvider, frmMessageBox는 실제 화면에 표시되는 Windows Form을 상속받는 클래스를 나타낸다. frmMessageBox 클래스는 메시지를 나타내고, frmSensorMemoryView 클래스는 센서정보 메모리 뷰어 기능을 담당한다. frmServiceProvider 클래스는 서비스 구동 제어, 서비스 공급자 상태 뷰어, 서비스 로그 뷰어의 기능을 담당한다.

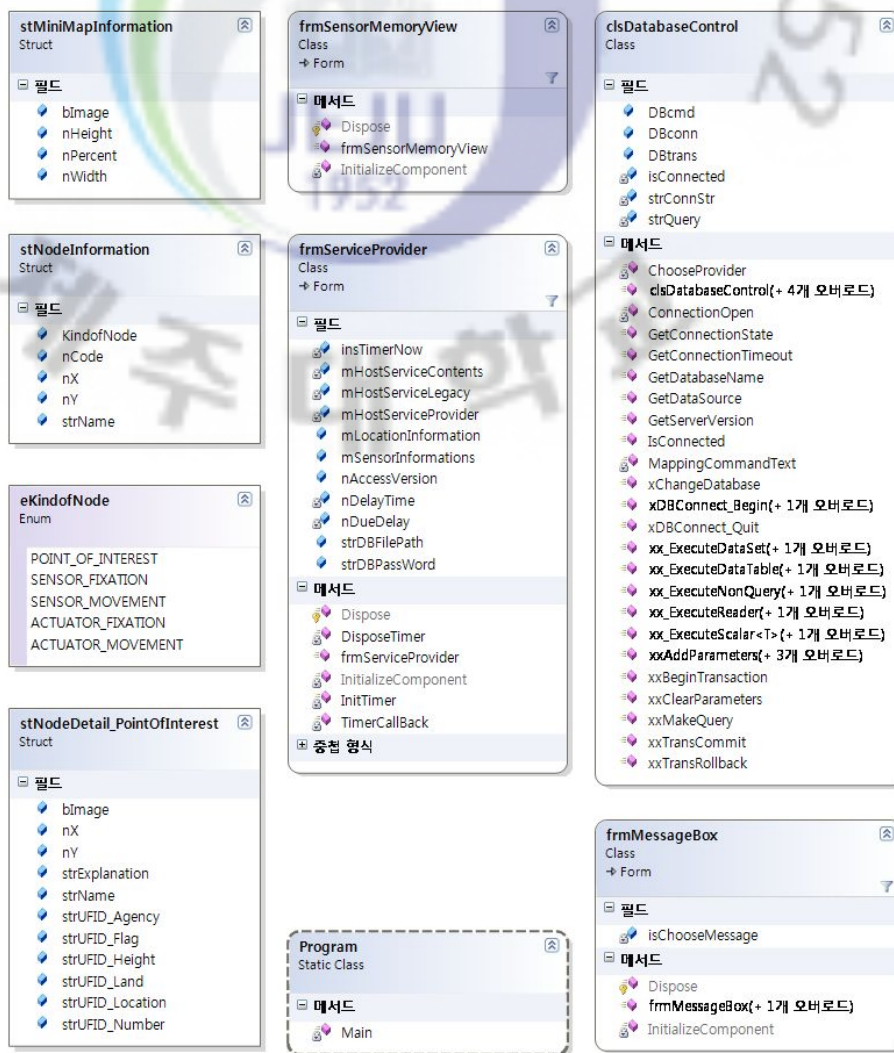


그림 34. 서비스 공급자의 클래스 다이어그램 2

서비스 공급자의 실행과정은 그림 35와 같다. 서비스 공급자의 프로그램 구동은 Click Once 를 통해 배포된다. 공급자는 서비스 공급자 메인화면과 센서정보 메모리 뷰어 화면으로 나누어진다. 서비스 공급자 메인화면에서 센서정보 메모리 뷰어를 선택하면 센서정보 메모리 뷰어 화면이 나타난다. 가상의 센싱 정보 생성은 센서 네트워크 플랫폼에서 서비스 공급자로 가상의 센싱 정보를 제공하기 위한 프로그램 이고, 센서의 인증 코드와 센서 분류를 기초로 센싱 정보를 전송하는 역할을 수행 한다.

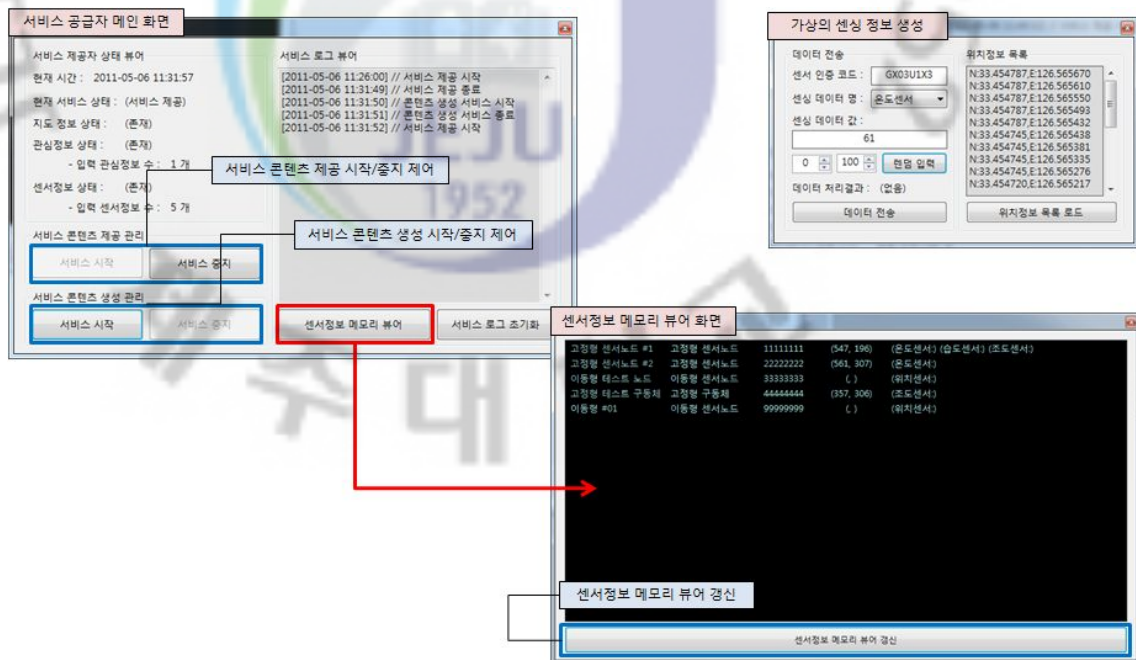


그림 35. 서비스 공급자의 실행과정

그림 36의 서비스 공급자 메인화면은 서비스 공급자 상태 뷰어(현재 서비스 상태, 지도 정보 상태, POI 상태, 센서정보 상태), 서비스 콘텐츠 제공 관리, 서비스 콘텐츠 생성 관리, 서비스 로그 뷰어, 센서정보 메모리 뷰어를 관리한다.

그림 37의 센서정보 메모리 뷰어 화면은 센서정보 메모리 뷰어(센서 명, 센서종류, 인증코드, 좌표값, 센싱정보 목록) 및 갱신을 모니터링 한다.



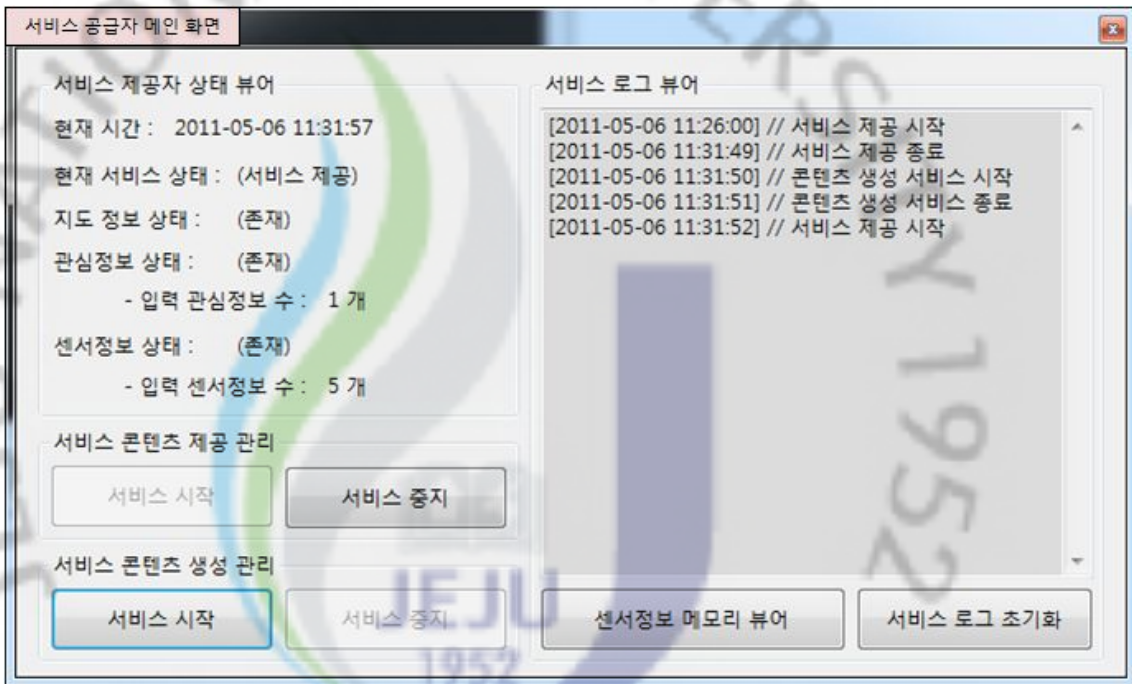


그림 36. 서비스 공급자 메인 실행화면

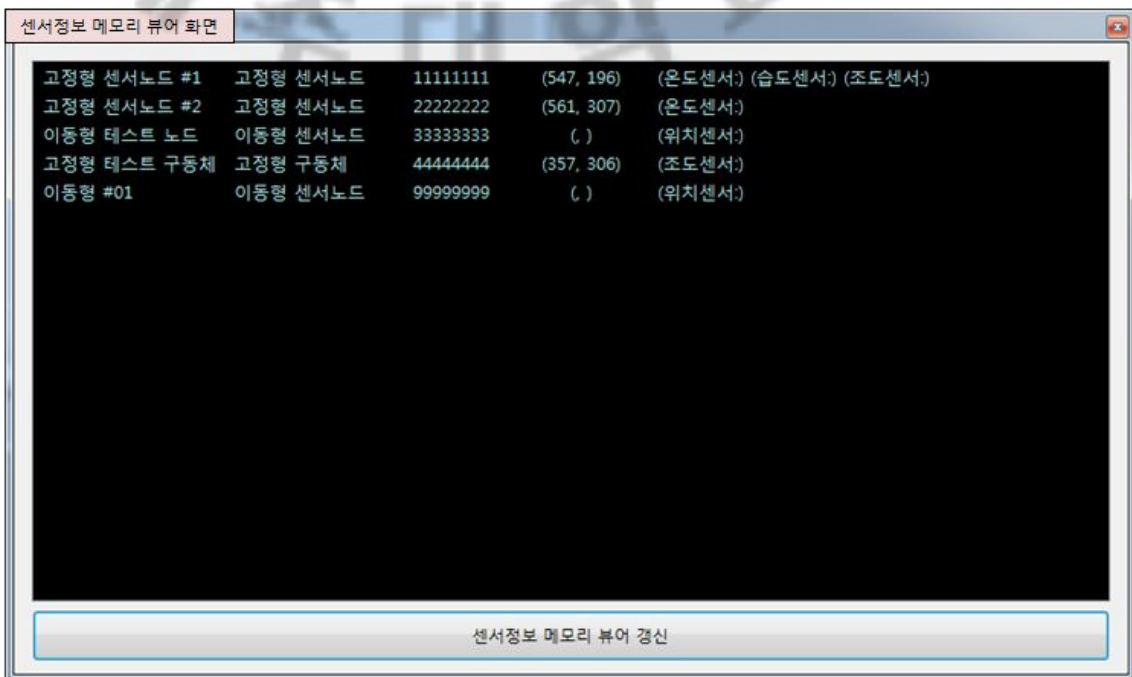


그림 37. 센서정보 메모리 뷰어 실행화면



### 3. COST 설계 및 구현

#### 1) COST 주요기능

기능 분류	주요 기능	세부 구성 요소
서비스 관리정보	서비스 공급자 정보 생성 및 관리	서비스 공급자 정보 관리
		지역 정보 뷰어
	서비스 콘텐츠 전송 관리	콘텐츠 정보 관리
		서비스 공급자 등록
		서비스 콘텐츠 등록
	전체 서비스 콘텐츠 상태 뷰어	
	지도정보 생성 및 관리	지도정보 뷰어
		미니지도 정보 관리
		원본 지도정보 분할
		분할지도 정보 가공
		지도정보 저장 관리
		저장 로그 뷰어
	기타 정보 관리	위치정보 매핑 관리
		지도정보 뷰어
		기초정보 뷰어
노드 정보 관리	센서 분류/타입 관리	
	구동체 정보 관리	
	센서정보 관리	
	POI 관리	
프로그램 배포관리	서비스 공급자를 위한 운영 프로그램 송신부	

표 5. COST의 주요기능

표 5는 COST의 주요기능을 나타낸다. COST의 기능은 서비스 관리정보와 프로그램 배포관리로 분류된다. 서비스 관리정보의 주요 기능은 서비스 공급자 정보 생성 및 관리, 서비스 콘텐츠 전송 관리, 전체 서비스 콘텐츠 상태 뷰어, 지도정보 생성 및 관리, 기타 정보 관리, 노드 정보 관리가 있다. 프로그램 배포관리의 주요 기능

은 서비스 공급자를 위한 운영 프로그램 송신부가 있다. COST는 콘텐츠를 생성하고 서비스 공급자를 위한 운영 프로그램과 서비스 콘텐츠를 전송하며, 서비스 저장소에 일괄적으로 서비스 공급자의 정보를 등록한다. 각각의 주요 기능은 그림 38에서 설명한다.

## 2) COST 설계

COST를 구성하는 세부항목으로는 서비스 공급자 정보 생성 및 관리, 서비스 콘텐츠 전송관리, 지도정보 생성 및 관리, 기타정보 관리, 노드 정보 관리, 전체 서비스 콘텐츠 상태 뷰어, XML Configuration, 데이터베이스로 구성되며, 서비스 공급자를 위한 운영 프로그램을 배포하는 기능을 수행한다.

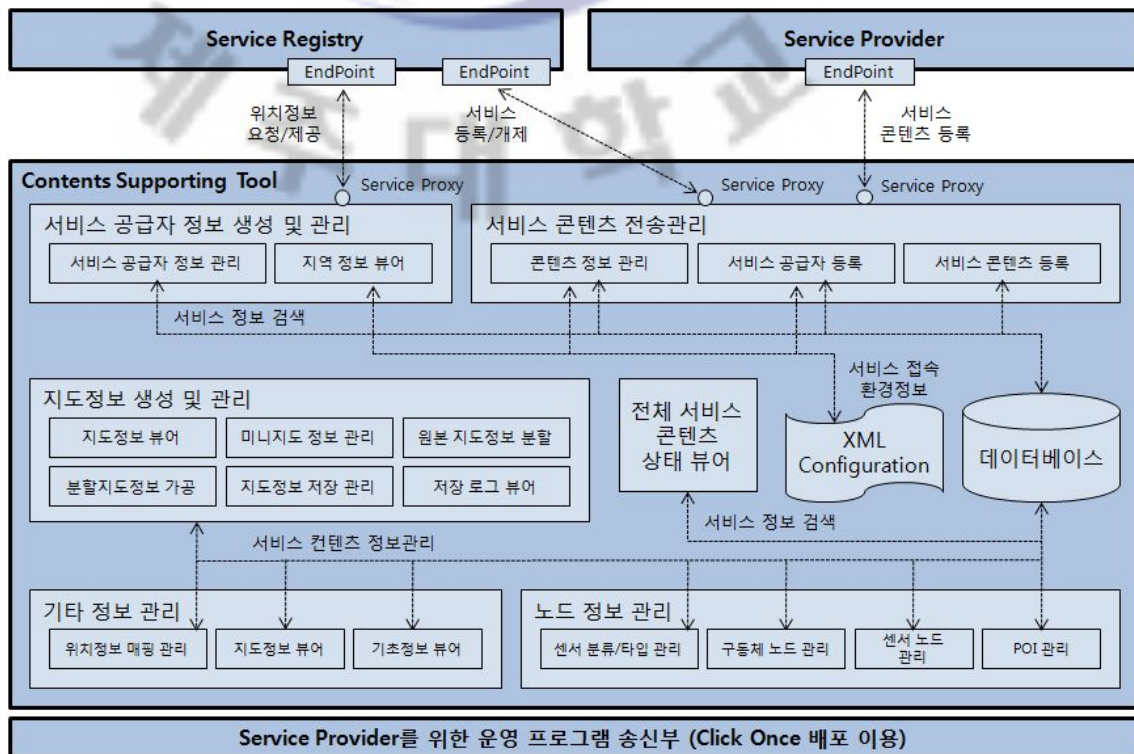


그림 38. COST 구성도

그림 38은 COST 구성도를 나타낸다. 서비스 공급자 정보 생성 및 관리는 서비스

공급자 정보를 관리하고 서비스 저장소에서 위치정보를 요청하여 지역정보를 나타낸다. 서비스 콘텐츠 전송관리는 생성된 데이터베이스를 기초로 콘텐츠 정보를 관리하고 서비스 공급자 등록 및 서비스 콘텐츠 등록을 한다. 지도정보 생성 및 관리는 지도정보 뷰어, 미니지도 정보 관리하고 원본 지도정보를 분할, 가공, 저장 및 관리하며 저장로그 뷰어의 기능이 있다. 기타정보 관리는 위치정보의 매핑 관리 및 지도정보 뷰어, 기초정보 뷰어의 기능을 한다. 노드 정보 관리는 센서 분류/타입 관리, 구동체 정보 관리, 센서정보 관리, POI 관리를 한다. 전체 서비스 콘텐츠 상태 뷰어는 데이터베이스에 기록된 전체 콘텐츠의 상태를 모니터링한다. XML Configuration은 서비스 접속정보를 포함한다. 데이터베이스는 지도정보 및 위치정보 매핑, POI노드, 센서노드, 구동체노드 등의 정보를 기록한다.

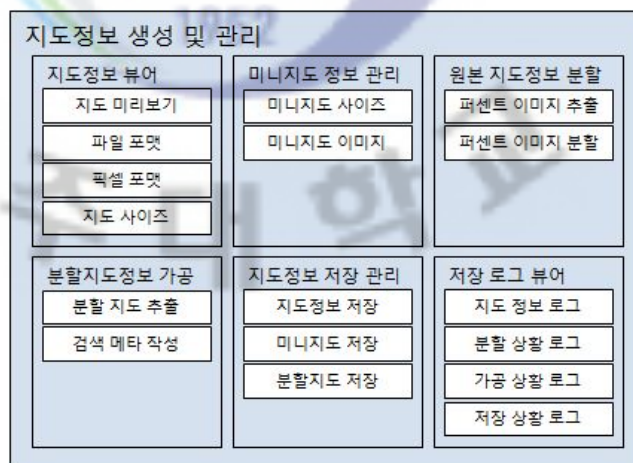


그림 39. 지도정보 생성 및 관리 모듈

그림 39는 지도정보 생성 및 관리 모듈이다. 해당 모듈에서 지도정보 뷰어 기능은 원본 지도 정보를 로드하고 지도 정보(파일 포맷, 픽셀 포맷, 원본 이미지 사이즈, 원본 이미지 미리보기)를 표시한다. 미니지도 정보 관리 기능은 입력된 미니지도 사이즈를 기초로 미니지도 제작을 관리한다. 원본 지도정보 분할 기능은 원본 지도 이미지를 100부터 10까지 퍼센트로 분류하여 각각의 이미지를 분할한다. 분할지도정보 가공 기능은 분할된 각각의 이미지에 검색을 위한 메타정보를 포함하여 가공하는 과정이다. 지도정보 저장 관리 기능은 가공이 완료된 지도정보를 데이터베이스

스에 일괄적으로 저장하는 역할을 수행한다. 저장 로그 뷰어 기능은 원본 지도정보 및 미니지도 정보의 저장로그와 가공처리 된 지도정보를 저장하는 과정을 모니터링 할 수 있다.

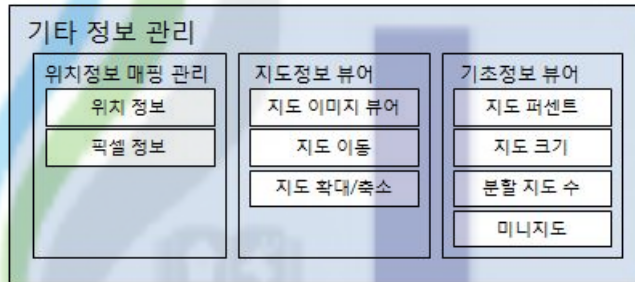


그림 40. 기타 정보 관리 모듈

그림 40은 기타 정보 관리 모듈이다. 해당 모듈에서 위치정보 매핑 관리 기능은 제작 된 픽셀 좌표 기반의 지도 이미지와 실제 위치정보(위도/경도)를 매핑하기 위해 두 점의 기본 위치좌표와 기준점을 설정하는 기능이다. 지도정보 뷰어 기능은 원본 지도 이미지를 표시하며, 편리한 정보 입력을 위해 지도 이동, 확대/축소, 좌표 선택의 역할을 지원한다. 기초정보 뷰어 기능은 위치매핑 정보(위치1, 위치2, 기준점), 기초 지도정보, 부가정보(지도 분할영역, POI, 센서/구동체정보), 선택 좌표값을 표시한다.



그림 41. 노드 정보 관리 모듈

그림 41은 노드 정보 관리 모듈이다. 해당 모듈에서 센서 분류/타입 관리 기능은 동적인 센서정보의 구성을 위해 센서 분류(온도센서, 습도센서, 조도센서 등)의 정



보를 관리하고 각각의 센서 타입(고정형 센서, 이동형 센서, 고정형 구동체 등)을 관리한다. 구동체 정보 관리 및 센서정보 관리 기능은 해당 노드의 접근을 위한 인증키, 위치정보, 장치 명, 장치 타입(고정형, 이동형 등), 장치 종류(온도, 습도, 전력 등), 장치 설명을 관리하고 서비스 공급자를 위해 SensorML의 기초정보를 생성하는 역할을 수행한다. POI 관리 기능은 POI 명, 위치 정보, 위치 설명, 연관 이미지, UFID 정보를 관리하는 역할을 수행한다.

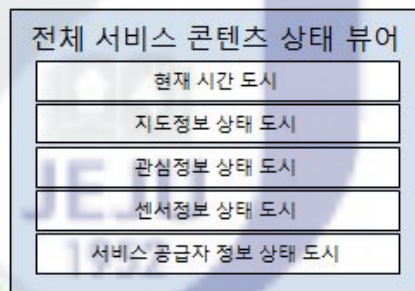


그림 42. 전체 서비스 콘텐츠 상태 뷰어 모듈

그림 42는 전체 서비스 콘텐츠 상태 뷰어 모듈이다. 해당 모듈은 생성된 콘텐츠 데이터베이스 정보의 상태를 로드하여 전체적인 상태를 모니터링 한다. 표시되는 콘텐츠 상태 정보는 현재 시간, 지도 정보 상태, 원본 지도 크기, 미니 지도 크기, 초기 위치정보 유무, POI 상태, 입력 POI 수, 센서정보 상태, 전체 입력 센서정보 수, 고정형 센서정보 수, 이동형 센서정보 수, 서비스 공급자 정보 상태가 있다.

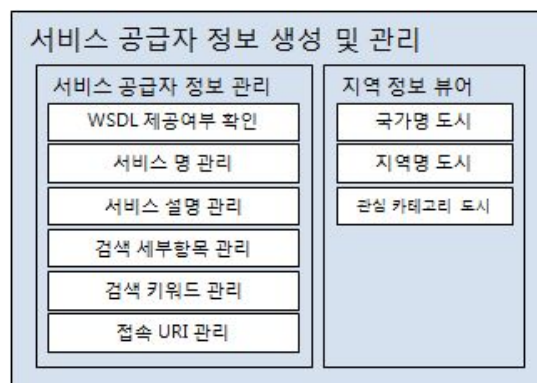


그림 43. 서비스 공급자 정보 생성 및 관리 모듈



그림 43은 서비스 공급자 정보 생성 및 관리 모듈이다. 해당 모듈에서 서비스 공급자 정보 관리 기능은 검색을 위한 공급자 정보를 관리하며 지역정보, 서비스 명, 서비스 설명, 검색 세부항목, 세부항목 설명, 검색 키워드, 접속 URI, WSDL 제공여부의 정보를 포함한다. 지역 정보 뷰어 기능은 서비스 저장소에서 제공하는 국가명, 지역명, 관심 분류의 정보를 표시하는 역할을 한다.

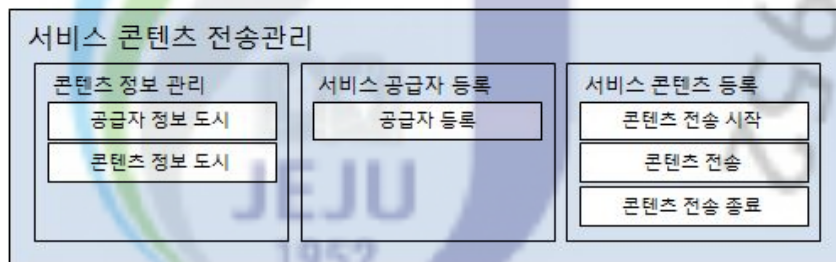


그림 44. 서비스 콘텐츠 전송관리 모듈

그림 44는 서비스 콘텐츠 전송관리 모듈이다. 해당 모듈에서 콘텐츠 정보 관리 기능은 COST가 관리하는 서비스 공급자 및 콘텐츠 정보의 상태를 표시하고 관리하는 역할을 한다. 서비스 공급자 등록 기능은 서비스 저장소로 해당 서비스 공급자 정보를 등록하는 역할이다. 서비스 콘텐츠 등록 기능은 COST에서 생성된 서비스 콘텐츠 정보를 해당 서비스 공급자에게 전송하는 역할을 수행한다.

그림 45는 COST의 시퀀스 다이어그램이다. COST안의 내부 모듈이 서비스 저장소와 서비스 공급자의 내부 처리과정을 나타낸다.

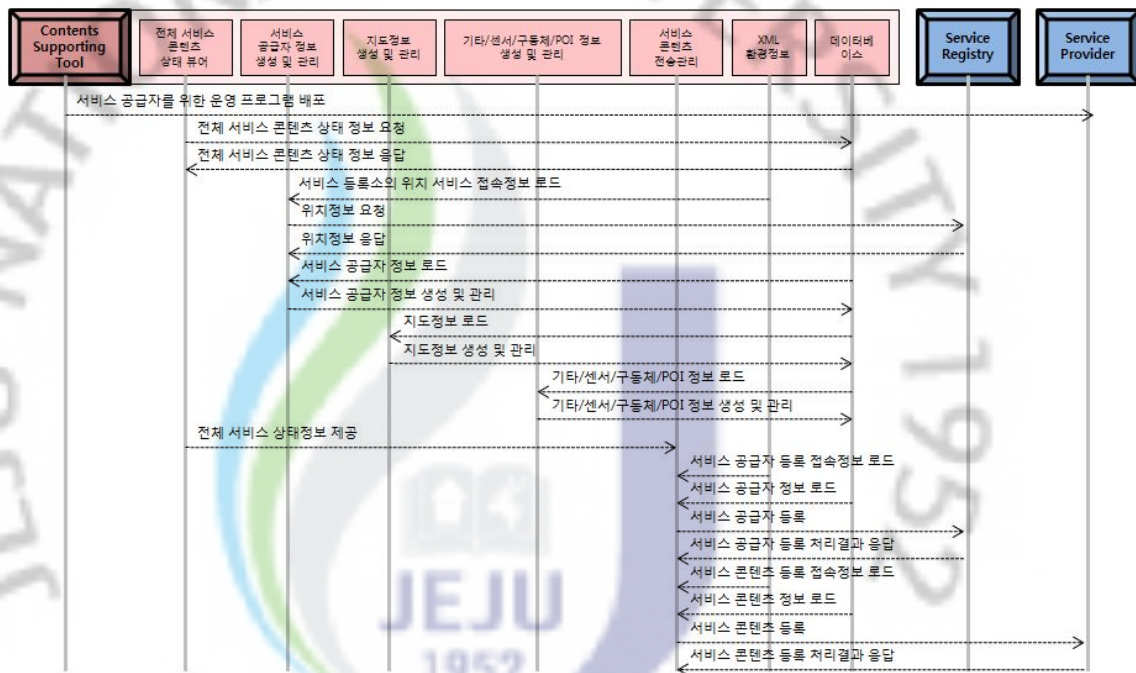


그림 45. COST 시퀀스 다이어그램

COST는 서비스 공급자를 위한 운영 프로그램을 Click Once 배포를 통해 전송한다. 전체 서비스 콘텐츠 상태 뷰어는 데이터베이스에서 전체 서비스 상태 정보를 요청하고 응답받는다. 서비스 공급자 정보 생성 및 관리에서 XML 환경정보를 통해 서비스 저장소의 위치 서비스 접속 정보를 로드하고 서비스 저장소에 위치정보를 요청하고 응답받아 표시한다. 서비스 공급자 정보 생성 및 관리는 데이터베이스에서 서비스 제공자 정보를 로드하고 공급자에 대한 정보를 생성하고 관리한다. 지도 정보 생성 및 관리 모듈은 데이터베이스에서 지도정보를 로드하고 지도정보에 대해 생성 및 관리한다. 지도정보가 생성 된 후 기타/센서/구동체/POI 정보 생성 및 관리에서 데이터베이스에 해당 노드의 정보를 생성 및 관리한다. 서비스 콘텐츠 전송관리는 전체 서비스 콘텐츠 상태 뷰어에서 전체 서비스 상태 정보를 제공받는다. 서비스 콘텐츠 전송관리 XML 환경정보에서 서비스 공급자 등록 접속정보를 로드하고 데이터베이스안의 서비스 공급자 정보를 서비스 저장소에 등록한다. 또한 XML 환경정보에서 서비스 콘텐츠 등록 접속정보를 로드하고 데이터베이스안의 콘텐츠 정보를 서비스 공급자에게 등록하는 역할을 한다.

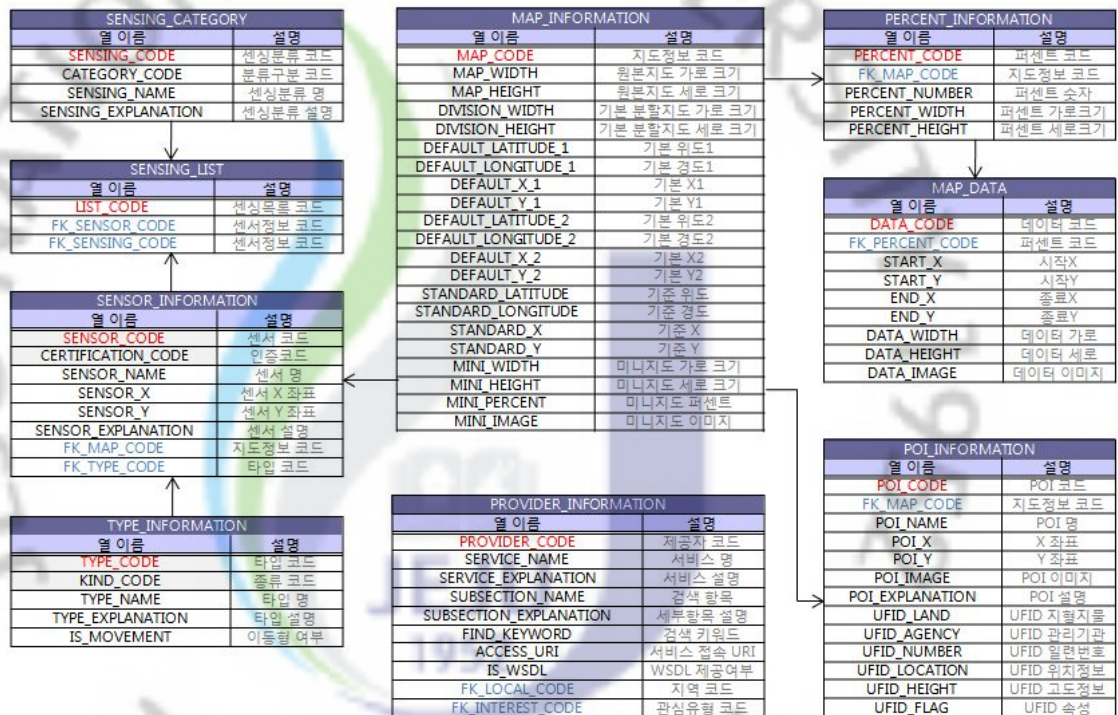


그림 46. COST 데이터베이스 객체 관계 다이어그램

그림 46은 COST의 데이터베이스 객체 관계 다이어그램을 나타낸다. 데이터베이스의 각각의 테이블은 MAP\_INFORMATION(지도정보), SENSING\_CATEGORY(센싱분류), SENSING\_LIST(센싱목록), SENSOR\_INFORMATION(센서정보), TYPE\_INFORMATION(타입설명), PERCENT\_INFORMATION(퍼센트정보), MAP\_DATA(분할지도 데이터), POI\_INFORMATION(관심정보), PROVIDER\_INFORMATION(공급자 정보)로 구성된다. MAP\_INFORMATION 테이블은 원본지도 데이터정보, 위치매핑 정보, 미니지도 정보를 포함한다. SENSING\_CATEGORY 테이블은 센싱분류 정보를 하고 SENSOR\_INFORMATION 테이블은 센서의 픽셀좌표 및 센서정보를 포함한다. TYPE\_INFORMATION 테이블은 각각의 센서에 대한 정보를 포함한다. SENSING\_LIST 테이블은 동적으로 하나의 센서에 대한 동적인 센싱분류 목록을 작성하기 위한 정보를 포함한다. PERCENT\_INFORMATION 테이블은 각각의 분할지도 정보에 대한 퍼센트 정보를 포함한다. MAP\_DATA 테이블은 분할된 지도정보를 포함한다. POI\_INFORMATION 테이블은 POI 노드에 대한 픽셀좌표와 UFID정보를 포함한다. PROVIDER\_INFORMATION 테이블은 서비스 공급자에 대

한 정보를 포함한다.

그림 47은 지도 및 노드정보의 생성을 나타낸다. 지도정보를 분할하고 가공하여 기록하고, 원본 이미지 정보에서 노드를 매핑하고 세부정보를 포함하여 기록하는 과정을 설명한다.

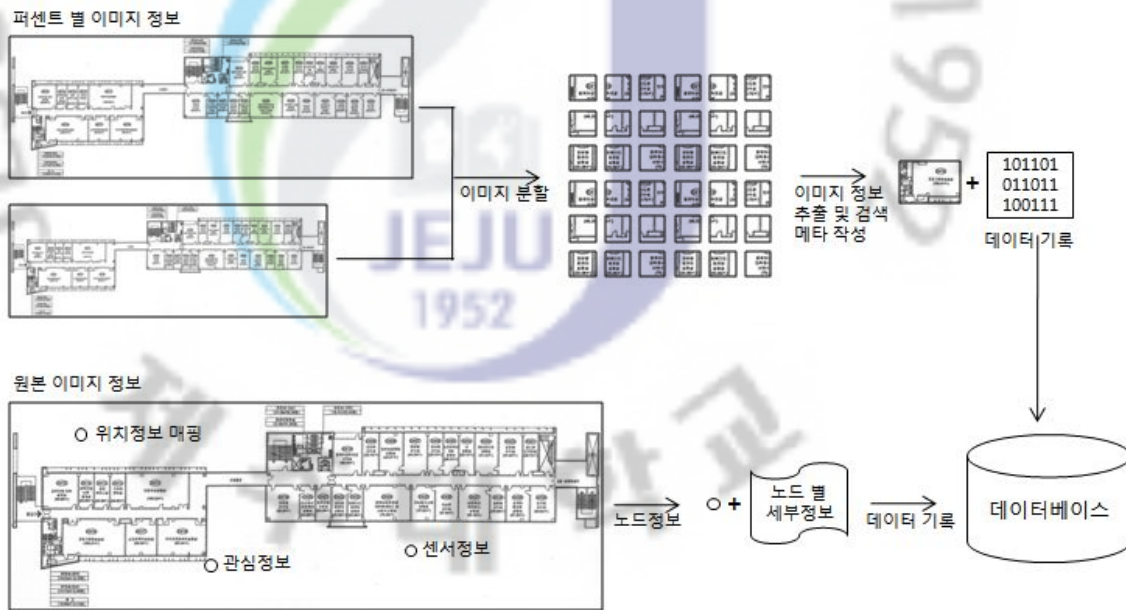


그림 47. 지도 및 노드정보 생성 개요

지도정보의 생성은 원본 이미지를 기초로 퍼센트 별로 이미지를 분류하고 각각의 이미지를 분할한다. 분할 된 이미지 정보를 추출하고 검색을 위한 메타 데이터를 작성한 후 데이터베이스에 데이터를 기록한다. 노드정보의 생성은 원본 이미지를 기초로 각각의 노드정보를 픽셀좌표로 매핑하고 해당노드와 노드 별 세부정보를 추가하여 데이터베이스에 기록한다.

각각의 정보 생성에 대한 세부적인 생성과정은 다음과 같다.



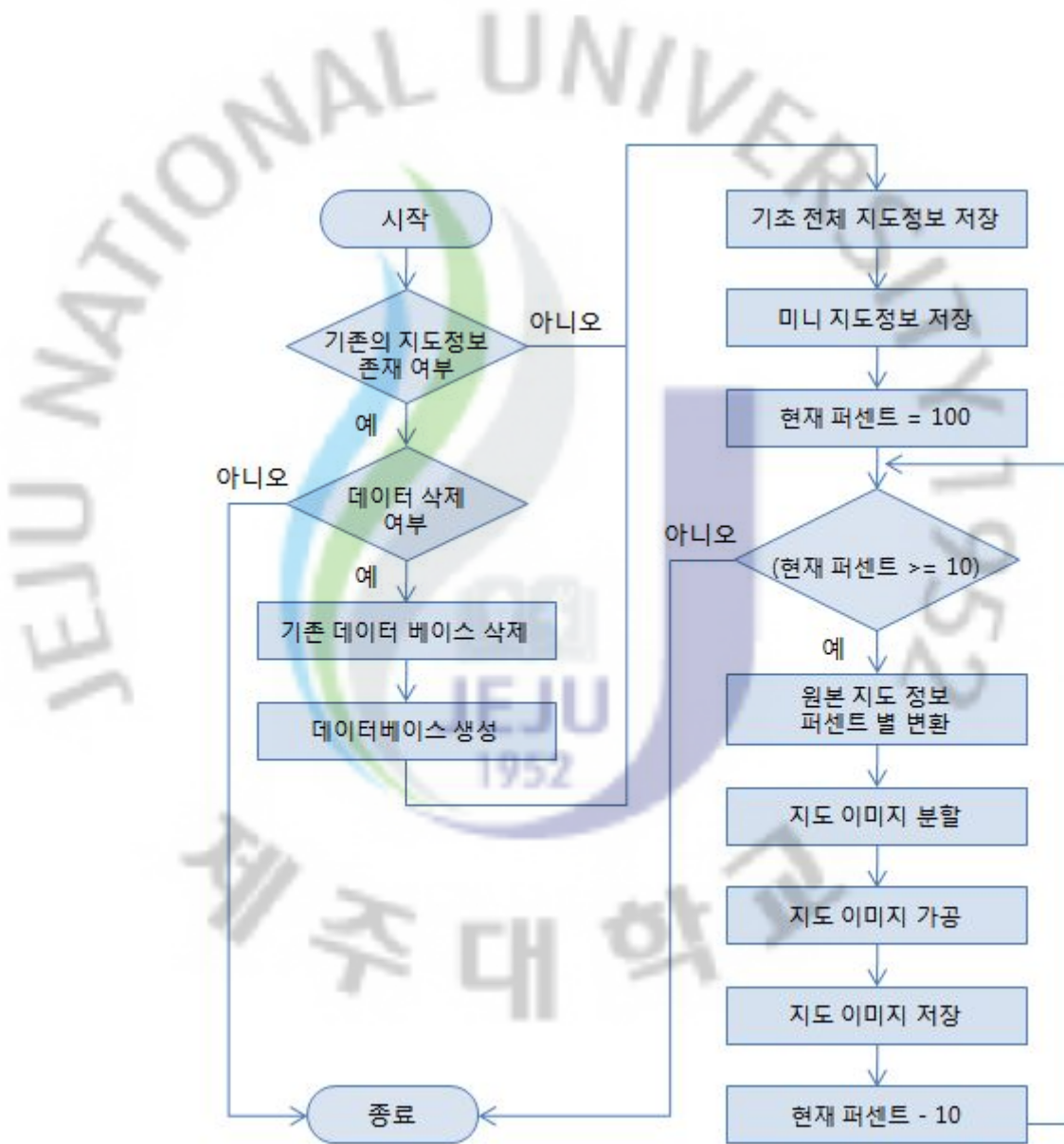


그림 48. 지도정보 생성 순서도

그림 48은 지도정보를 생성하는 과정에 대한 순서도를 나타낸다. 지도정보를 생성하기 위해 기존의 지도정보 존재 여부를 확인하고 존재 한다면 데이터 삭제 여부를 확인하여 삭제 확인을 하고 기존 데이터베이스를 삭제하고 데이터베이스를 생성한다. 기초 전체 지도를 로드하고 정보를 저장한 후 미니 지도정보를 저장한다. 100 퍼센트부터 10 퍼센트 까지의 이미지를 대상으로 이미지 분할, 가공, 저장을 한다.

그림 49는 노트정보를 생성하는 과정에 대한 순서도를 나타낸다.



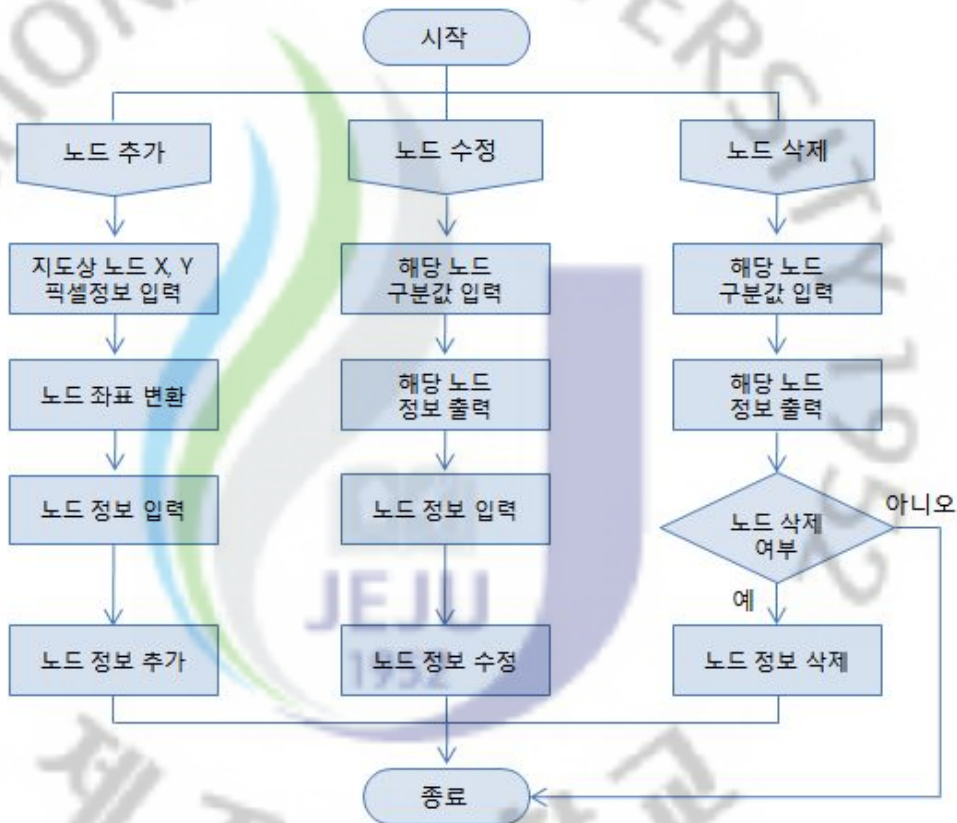


그림 49. 노드정보 생성 순서도

노드정보를 생성하기위해 노드정보를 추가, 수정, 삭제하는 과정을 수행한다. 노드를 추가하기 위해서 지도상 노드의 픽셀정보를 입력하고 노드의 좌표를 변환하고 세부 노드정보를 입력한 후 기록을 한다. 노드를 수정은 해당 노드의 구분값을 입력하고 해당 노드 정보를 출력하고 세부 노드정보를 입력한 후 수정을 한다. 노드를 삭제는 해당 노드의 구분값 입력을 기초로 해당 노드 정보를 출력하고 노드삭제 여부를 확인한 후 삭제를 한다.

### 3) COST 구현

본 논문에서 제안하는 센서웹에서 COST 연구의 구현은 다음과 같다.

그림 50, 51은 COST의 클래스 다이어그램을 나타낸다.

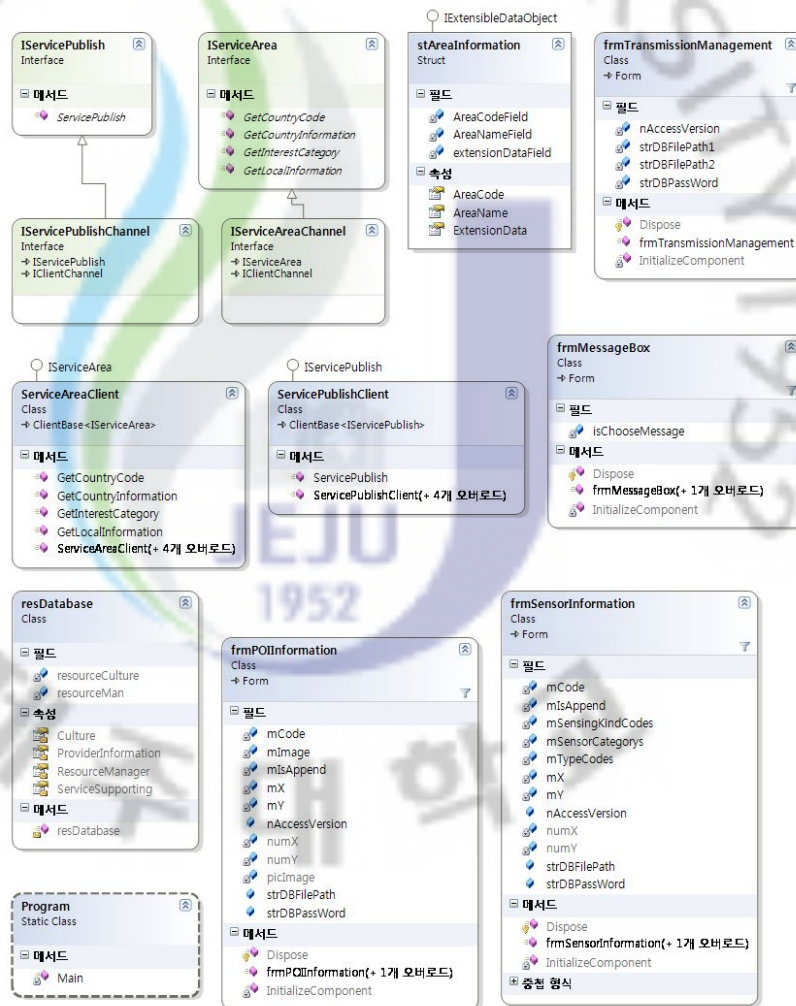


그림 50. COST의 클래스 다이어그램 1

IServicePublish, IServiceArea는 인터페이스를 나타내고, ServicePublishClient, ServiceAreaClient는 인터페이스를 상속받는 클래스를 나타낸다. clsDatabaseControl은 데이터베이스를 제어하기 위한 클래스를 나타낸다. stAreaInformation는 전역으로 사용하는 구조체를 나타낸다. frmTransmissionManagement, frmMessageBox, frmPOIInformation, frmSensorInformation, frmProviderManagement, frmMovementNode, frmTypeCategory, frmMapManagement, frmSensingCategory, frmLocationInformation, frmServiceSupportingTool, frmOtherManagement 실제 화면에 표시되는 Windows Form을 상속받는 클래스를 나타낸다. frmMessageBox 클래스는 메시지를

나타내고, frmTransmissionManagement 클래스는 서비스 콘텐츠 전송 기능을 담당한다. 노드 정보 관리 기능에서 frmPOIInformation 클래스는 POI 정보 관리, frmSensorInformation 클래스는 센서정보 관리, frmMovementNode 클래스는 모든 이동형 노드 관리, frmTypeCategory 클래스는 센서의 타입(고정형, 이동형 센서/구동체) 관리, frmSensingCategory 클래스는 센서의 분류(온도, 습도, 조도 등)를 관리한다. frmProviderManagement 클래스는 서비스 공급자 정보 생성 및 관리 기능을 담당한다. frmMapManagement 클래스는 지도정보 생성 및 관리 기능을 담당한다. frmOtherManagement 클래스는 기타정보 관리 기능을 담당하고, frmLocationInformation 클래스는 기타 정보 관리에서 위치정보 매핑 관리를 한다. frmServiceSupportingTool 클래스는 전체 서비스 콘텐츠 상태 뷰어 기능을 한다.

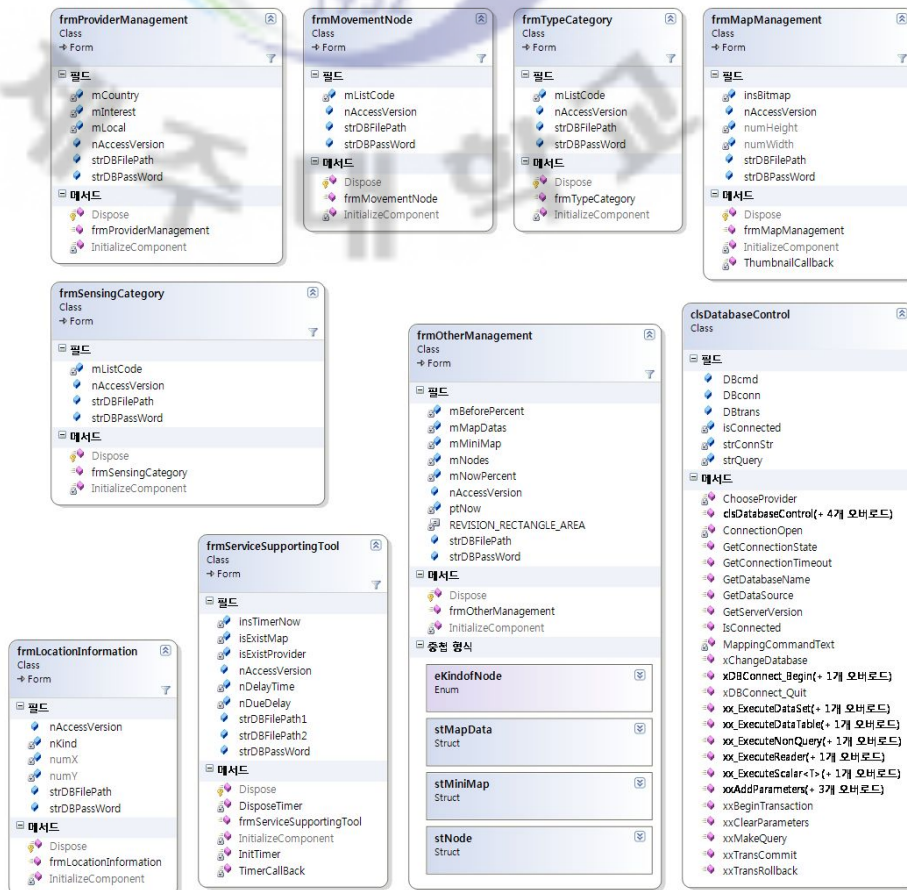


그림 51. COST의 클래스 다이어그램 2

COST의 실행과정은 그림 52와 같다. COST 메인화면, 기타정보 관리화면, 서비스 공급자 정보화면, 지도정보 관리화면, 서비스 콘텐츠 전송 관리로 나누어진다.

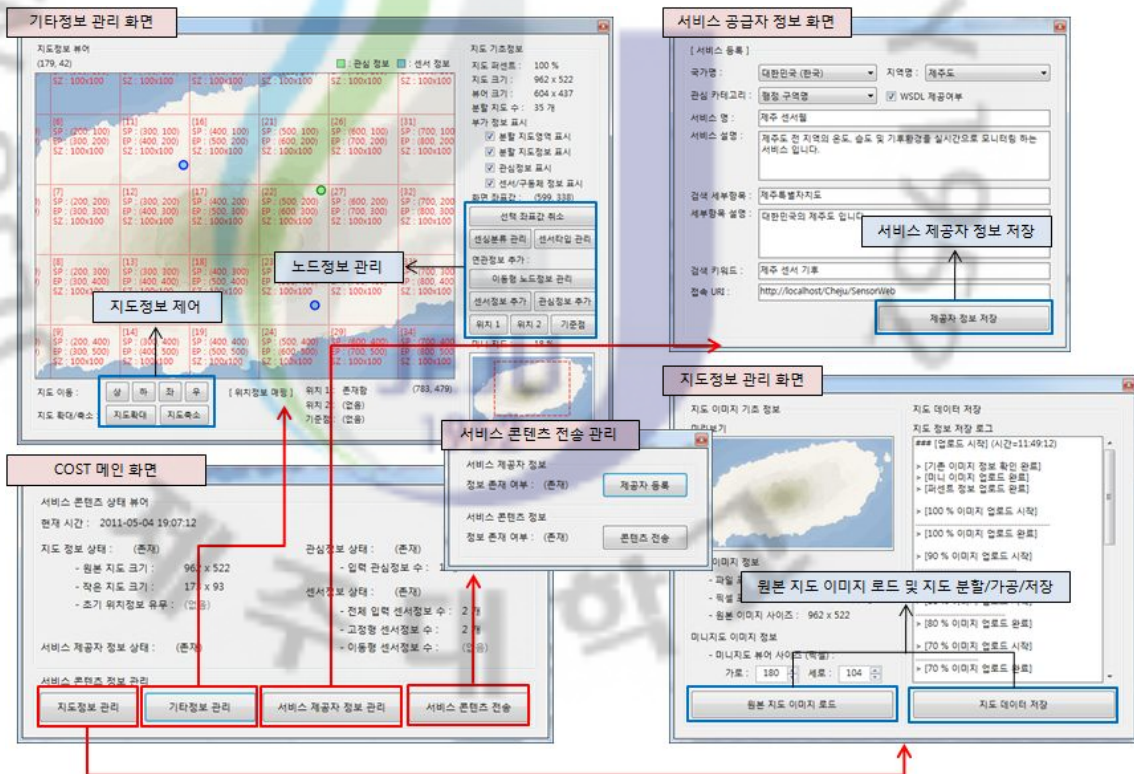


그림 52. COST의 실행과정

COST 메인화면은 서비스 콘텐츠 상태 모니터링, 지도정보 관리, 기타정보 관리, 서비스 공급자 정보 관리, 서비스 콘텐츠 전송을 관리한다. 지도정보 관리 화면은 원본 지도 이미지를 로드하고 정보를 표시하고 지도 데이터를 분할/가공/저장을 한다. 기타정보 관리화면은 지도정보를 표시, 지도뷰어를 제어(지도이동, 지도 확대/축소), 지도 기초정보를 확인, 노드정보 관리(센싱분류, 센서타입, 이동형 노드, 센서정보, POI, 위치, 기준점), 미니지도 뷰어의 역할을 수행한다. 서비스 공급자 정보 화면은 서비스 등록을 위해 위치정보 및 서비스 정보를 저장하는 역할을 한다. 서비스 콘텐츠 전송 관리는 서비스 공급자 및 콘텐츠 정보를 전송하는 역할을 수행한다. COST 메인화면에서 지도정보 관리를 선택하면 지도정보 관리 화면, 기타정보



관리를 선택하면 기타정보 화면, 서비스 공급자 정보 관리를 선택하면 서비스 공급자 화면, 서비스 콘텐츠 전송을 선택하면 서비스 콘텐츠 전송 관리 화면이 나타난다.

그림 53은 COST 메인 실행화면을 나타낸다. 화면 상단은 서비스 콘텐츠의 상태를 모니터링하며 현재시간, 지도정보 상태, POI 상태, 센서정보 상태, 서비스 공급자 정보 상태를 표시한다. 화면 하단은 서비스 콘텐츠 정보 관리 기능이며 지도정보 관리, 기타정보 관리, 서비스 제공자 정보 관리, 서비스 콘텐츠 전송을 관리한다.

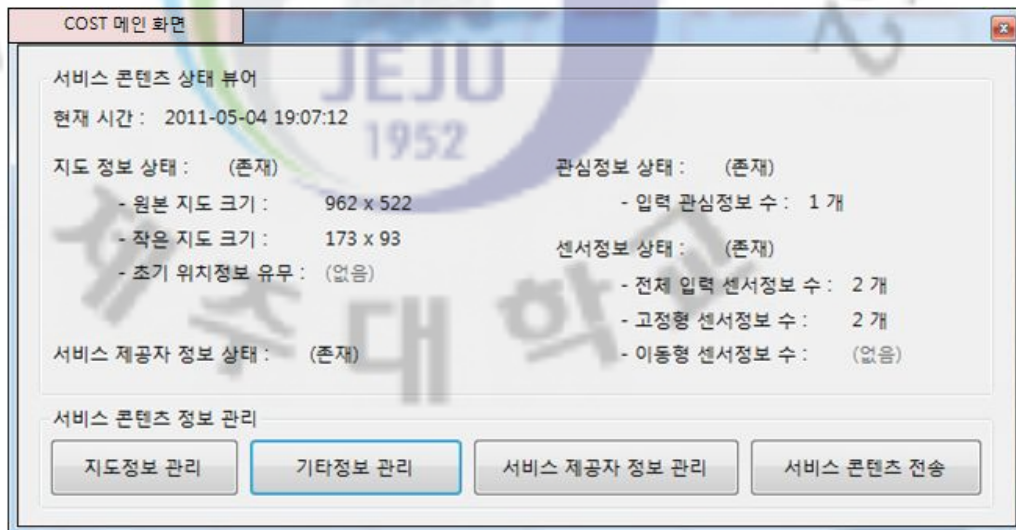


그림 53. COST 실행화면

그림 54는 지도정보 관리 실행화면을 나타낸다. 화면 좌측은 지도 이미지 기초정보 뷰어이며 원본 지도 이미지 로드, 미리보기, 지도 이미지 정보(파일 포맷, 픽셀 포맷, 원본 이미지 사이즈), 미니지도 이미지 정보를 표시한다. 화면 우측은 지도 데이터 저장의 로그를 나타내며 지도 데이터를 저장할 수 있는 기능과 지도 정보 저장을 로그로 표현한다.



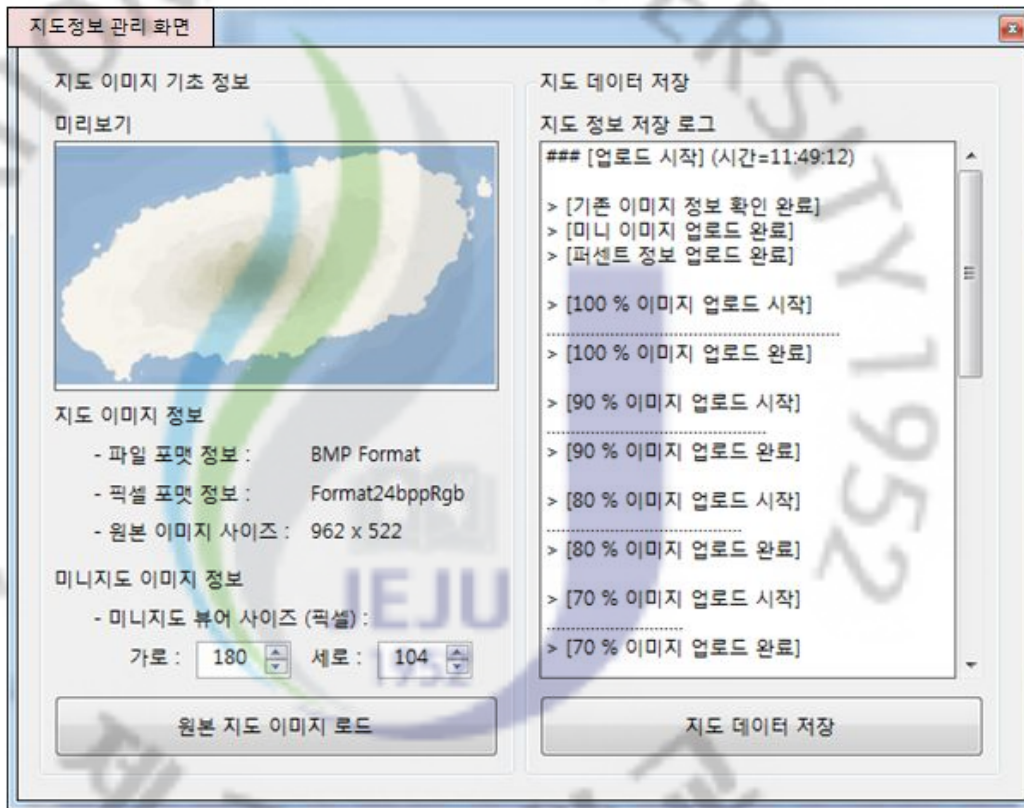


그림 54. 지도정보 관리 실행화면

그림 55는 기타정보 관리 실행화면을 나타낸다. 화면 좌측은 지도정보 뷰어이며 지도 이미지와 노드정보를 표현하고 지도이동, 지도 확대/축소, 원치정보 매핑 상태를 나타낸다. 화면 우측은 지도 기초정보 뷰어이며 지도 퍼센트, 지도 크기, 뷰어 크기, 분할 지도 수, 부가 정보 표시, 화면 좌표값, 미니지도 정보를 표시하고 노드 및 위치매핑 정보를 제어한다.

그림 56은 서비스 공급자 정보 및 서비스 콘텐츠 전송 관리 실행화면을 나타낸다. 서비스 공급자 정보 관리 실행화면에서는 국가명, 지역명, 관심분류, WSDL 제공여부, 서비스 명, 서비스 설명, 검색 세부항목, 세부항목 설명, 검색 키워드, 접속 URI 정보를 나타내고 관리한다. 서비스 콘텐츠 전송관리 실행화면에서는 서비스 공급자 정보의 존재여부, 서비스 콘텐츠 정보의 존재여부를 나타내고 공급자 등록 및 콘텐츠 전송의 역할을 수행한다.



그림 55. 기타정보 관리 실행화면

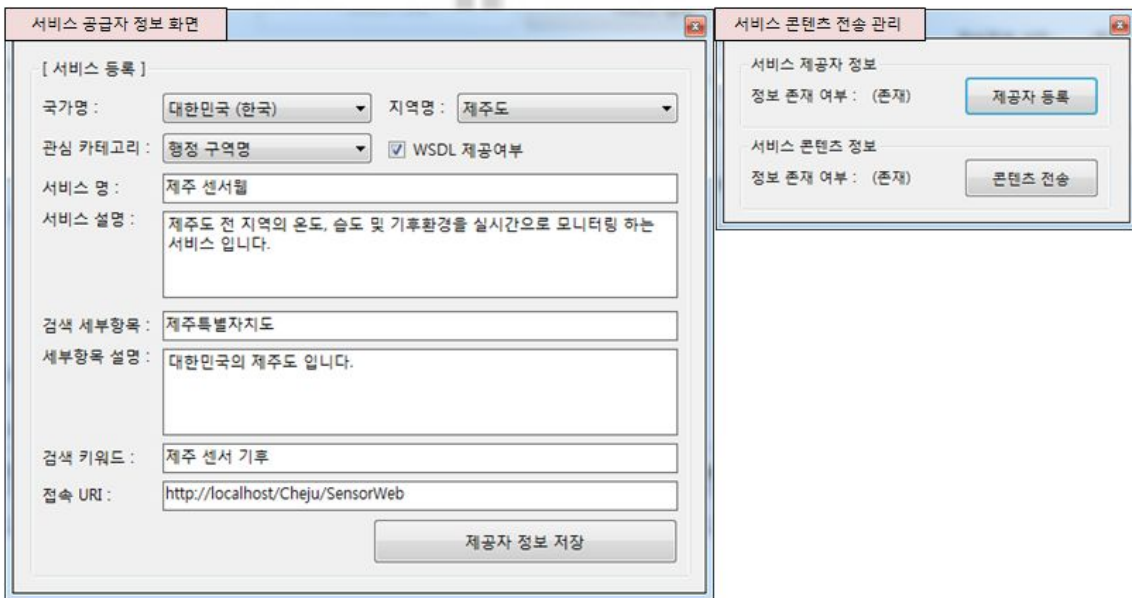


그림 56. 서비스 공급자 정보 및 서비스 콘텐츠 전송 관리 실행화면

## IV. 성능분석

### 1. 실험 환경

실험환경은 표 6과 같다. 실험은 COST와 서비스 요청자의 주요 기능에서 성능분석을 하였다. 모든 성능분석은 해당 주요모듈 별로 20회에 걸쳐 수행시간을 측정하고 분석하였다.

COST에서는 콘텐츠의 생성과 콘텐츠 전송이 수행되는 시간을 측정하였다. 콘텐츠 생성은 동일한 지도 이미지를 로드하고 분할 및 가공하여 데이터베이스에 기록하는 시간을 측정하였고, 콘텐츠 전송은 COST에서 생성한 데이터베이스를 서비스 공급자에게 전송하는 시간을 측정하였다. 또한 콘텐츠가 전송 되고 서비스 요청자에서 정상적으로 사용되는지 실험을 위해 서비스 요청자에서 서비스 검색과 콘텐츠 요청 및 응답이 수행되는 시간을 측정하였다. 서비스 검색은 서비스 요청자가 서비스 저장소에서 해당 서비스 공급자의 접속정보를 응답 받는 시간을 측정하였고, 콘텐츠 요청 및 응답은 해당 공급자에게 콘텐츠를 요청하고 응답받은 후 화면에 표현하는 시간을 측정하였다.

구분	세부내용
개발운영체제	Microsoft Windows 7
개발환경	.Net Framework 3.0 이상
개발언어	Visual Studio .Net 2008
DBMS	Microsoft Access 2007
하드웨어	CPU : Intel(R) Core(TM)2 Duo E8400 CPU @ 3.00GHz RAM : 3.25 GB DisPlay : NVIDIA GeForce 9600 GT

표 6. 실험 환경

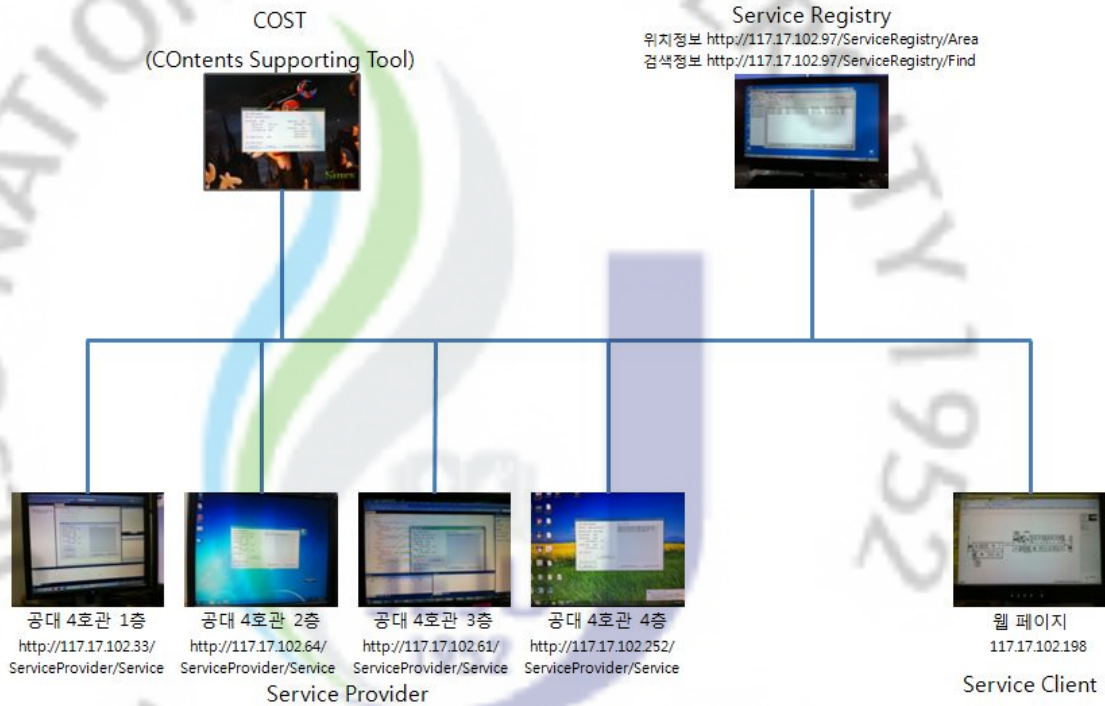


그림 57. 실험환경 네트워크 구성

실험환경의 네트워크 구성은 그림 57과 같다. 실험을 위해 COST를 구동하기 위한 데스크탑 1대, 서비스 저장소를 위한 서버 1대, 서비스 공급자를 위한 서버 4대, 서비스 요청자를 위한 노트북 1대를 사용하였다.

COST는 데스크탑에서 서비스 공급자를 위해 콘텐츠를 생성하고 전송하며, 서비스 저장소에 일괄적으로 서비스 공급자를 등록한다. 서비스 저장소는 서버에서 서비스 요청자의 검색을 지원하기 위해 동작한다. 서비스 공급자는 서버4대에 공과대학 4호관의 1층, 2층, 3층, 4층 콘텐츠를 제공하며 서비스 요청자에서 접속을 수락하고 해당 콘텐츠를 제공하는 역할을 수행한다. 서비스 요청자는 노트북에서 유/무선으로 서비스 저장소의 공급자 정보를 검색하고 해당하는 공급자의 콘텐츠를 사용하는 역할을 수행한다.

서비스 실시간 센싱정보를 제공하는 서비스 공급자를 위한 센서 네트워크 플랫폼은 그림 58과 같다.



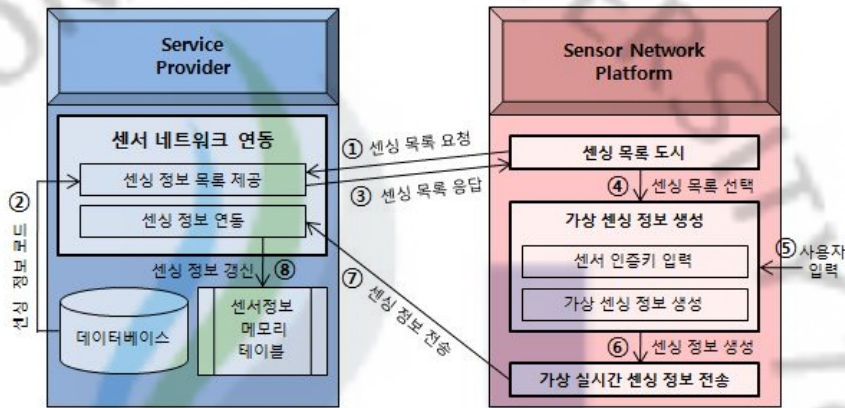


그림 58. 가상 센싱 정보 생성과정

센서 네트워크 플랫폼은 가상의 센싱 정보를 생성하고 전송하는 역할을 담당한다. 실험을 위해 서비스 공급자에게 가상의 실시간 센싱 정보를 전송할 수 있는 센서 네트워크 플랫폼을 구현하였다. 전체적인 처리 흐름은 ①에서 센서 네트워크 플랫폼의 센싱 목록 도시 모듈에서 서비스 공급자에게 센싱 목록을 요청한다. 서비스 공급자의 센싱 정보 목록 제공 기능에서는 ②와 같이 데이터베이스에서 센싱 정보 목록을 로드하고 ③에서 정보를 전송한다. ④에서 센서 플랫폼은 센싱 목록을 선택하고 ⑤와 같이 사용자 입력을 통해 센서의 인증키와 가상 센싱 정보를 생성한다. ⑥에서 생성된 센싱 정보 생성을 가상 실시간 센싱 정보 전송 모듈을 통해 ⑦과 같이 정보 전송을 한다. 서비스 공급자의 센싱 정보 연동 기능에서는 허용된 센싱 인증키를 확인하고 ⑧과 같이 센서정보 메모리 테이블에 정보를 갱신하여 최종적으로 서비스 요청자에게 서비스한다. 실제 센서 네트워크 플랫폼의 실행화면은 그림 59와 같다.

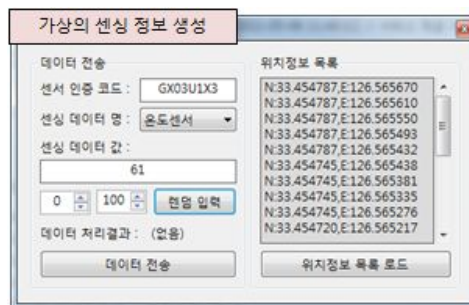


그림 59. 가상 센싱 정보 생성화면



## 2. 실험 결과

전체 COST 실험의 개념은 그림 60과 같으며, COST와 각각의 서비스 공급자, 요청자의 관계를 나타낸다.

COST는 Service Information Manager를 통해 해당 공급자의 정보를 서비스 저장소에 등록한다. Platform Manager를 통해 서비스 공급자의 프로그램을 배포하고, Contents Edit & Manager를 통해 콘텐츠를 생성하고 공급자에게 전송한다.

서비스 요청자는 센서웹 서비스를 제공받기 위해 서비스 저장소에 공급자 정보를 검색하고, 해당 서비스 공급자로 접속하고 콘텐츠를 제공받는다.

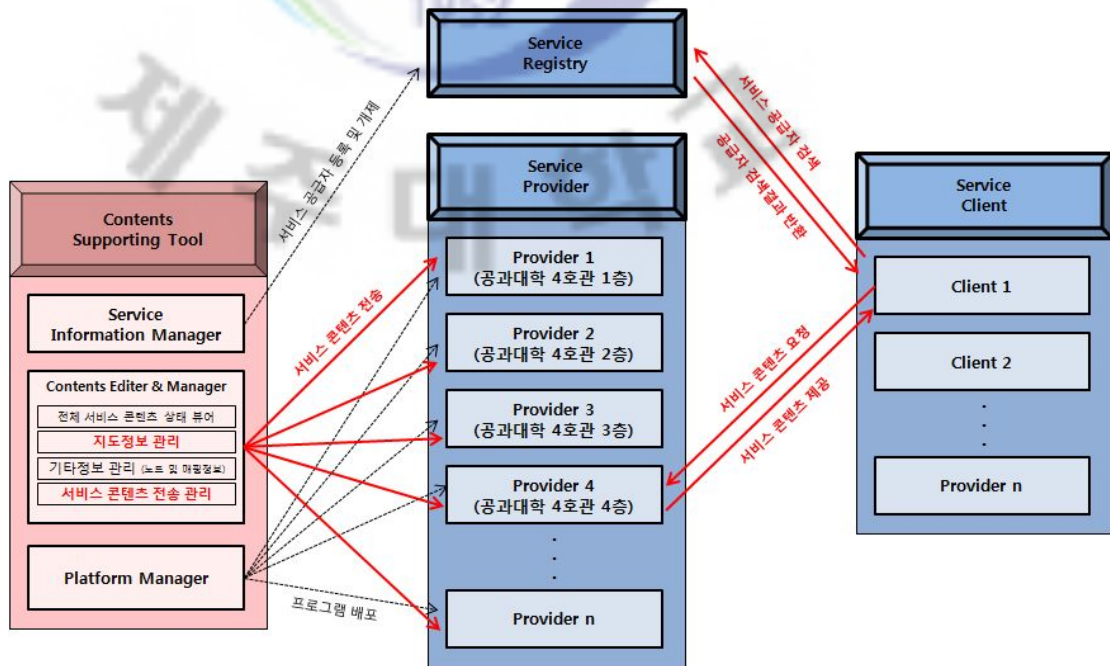


그림 60. COST 실험 개념도

실험에서 성능평가는 COST에서 서비스 공급자로 콘텐츠를 생성하고 전송하는 시간과 서비스 요청자가 서비스 저장소에서 공급자를 검색하는 시간, 서비스 요청자가 서비스 공급자에게 콘텐츠를 전송 받는 시간을 측정하였다.

## 1) 콘텐츠 생성 및 전송

콘텐츠 생성화면은 그림 61과 같다. 콘텐츠 생성은 COST에서 서비스 공급자를 위해 해당 콘텐츠를 생성하는 단계이다. 실험을 위해 지도 이미지는 PNG 포맷의 1024 x 417 사이즈의 이미지를 사용하였다. 지도 이미지를 로드하여 총 20회에 걸쳐 지도 데이터를 저장하였고 지도 데이터가 로드되고 퍼센트 별로 구분되어 분할 및 가공, 데이터베이스에 저장되는 수행 시간을 측정하였다.

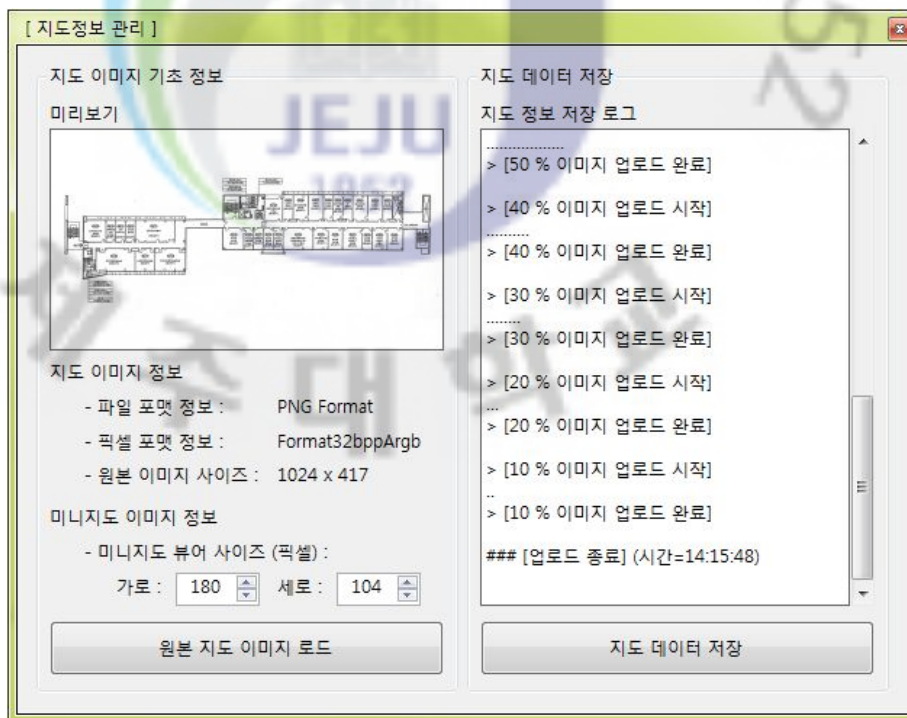


그림 61. 콘텐츠 생성

콘텐츠 생성에 대한 성능분석은 그림 62와 같다. 그래프의 가로축은 수행 횟수를 나타내고 세로축은 수행시간(초)을 나타낸다. 각각의 수행횟수 별로 30.22973, 35.10901, 32.78387, 39.24725, 31.61481, 32.44386, 30.59875, 30.92877, 32.61987, 31.62081, 35.14518, 33.15589, 31.96821, 30.69876, 32.46002, 33.87911, 31.56003, 30.75084, 32.08913, 31.35484 초의 수행시간이 측정되었고 최소 수행시간은 30.22973 초이고, 최대 수행시간은 39.24725 이다. 평균적으로 1024 x 417 사이즈의 지도 이미지를 가공하

고 저장하는데 걸리는 시간은 32.512937 초로 분석되었다.



그림 62. 콘텐츠 생성의 소요시간

콘텐츠 전송화면은 그림 63, 64와 같다. 콘텐츠 전송은 COST에서 서비스 공급자로 콘텐츠를 전송하는 단계이다. COST에서 콘텐츠 전송관리 기능을 통해 그림 58의 서비스 공급자에게 생성된 콘텐츠를 전송한다. 실험을 위해 전송되는 콘텐츠 데이터베이스의 크기는 1,884KB를 사용하였다. 콘텐츠 전송은 총 20회에 걸쳐 실험하였고 서비스 공급자에게 콘텐츠 전송시작, 전송, 완료하는 수행 시간을 측정하였다.

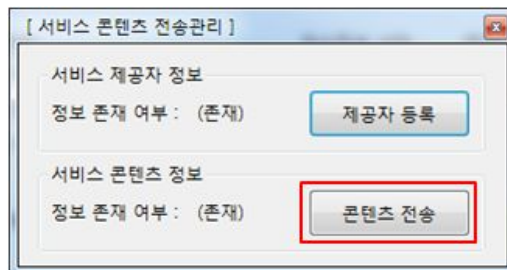


그림 63. 콘텐츠 전송

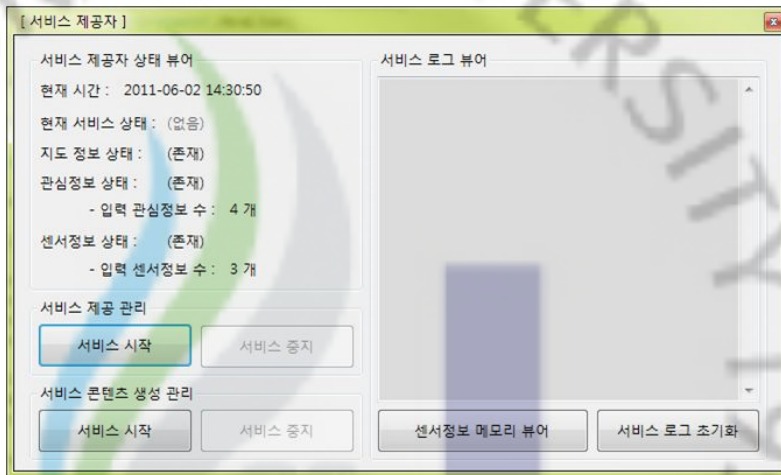


그림 64. 서비스 공급자

콘텐츠 전송의 성능분석은 그림 65와 같다. 그래프의 가로축은 수행 횟수를 나타내고 세로축은 수행시간(초)을 나타낸다. 각각의 수행횟수 별로 9.407537, 8.849506, 9.855564, 8.243471, 9.065518, 9.495543, 8.168468, 10.80762, 8.759501, 10.65061, 8.848621, 9.788014, 8.468301, 10.16876, 9.468661, 9.807611, 8.460017, 8.234751, 8.212604, 8.515401 초의 수행시간이 측정되었고 최소 수행시간은 8.168468 초이고, 최대 수행시간은 10.80762 이다. 평균적으로 1,884KB 크기의 콘텐츠 데이터베이스를 전송하는데 걸리는 시간은 9.16380395 초로 분석되었다.

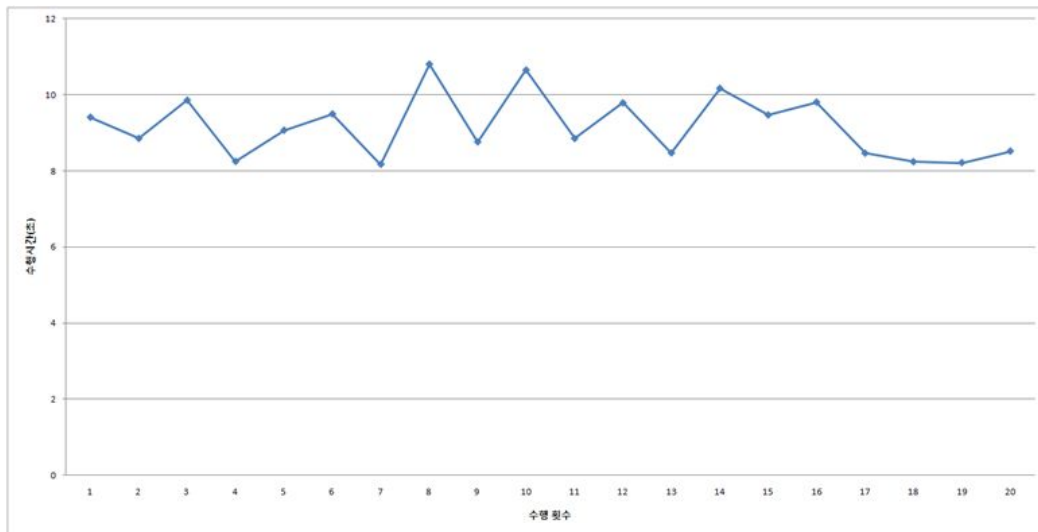


그림 65. 콘텐츠 전송의 소요시간

## 2) 서비스 공급자 검색 및 콘텐츠 제공

본 논문에서 제안하는 COST와 다양한 센서 네트워크로부터 실시간으로 수집되는 상황 데이터를 통합하고 처리한 정보를 제공하는 SOA 기반의 센서웹과의 연동을 실험하기 위해 서비스 요청자에서 서비스 공급자 검색 및 콘텐츠 제공을 실험하였다.

공급자 검색화면은 그림 66과 같다. 공급자 검색은 서비스 요청자에서 원하는 콘텐츠를 제공받기 위해 서비스 공급자를 검색하는 단계이다. 실험을 위해 임의의 공급자를 검색하여 총 20회에 걸쳐 서비스 공급자를 검색하였고 공급자 검색 응답에 소요되는 시간을 측정하였다.

**[ 서비스 레지스트리 ]**

지역 레지스트리 :

검색 레지스트리 :

**[ 서비스 검색 ]**

국가명 :       검색 세부항목 :

지역명 :       검색 키워드 :

관심 카테고리 :      

**서비스 검색 결과 :**

NUM	ServiceName	ServiceURI	WSDL
0	제주대학교 공과대학 4호관 1층	http://localhost/ServiceProvider/Service1	제공
1	제주대학교 공과대학 4호관 2층	http://localhost/ServiceProvider/Service2	제공
2	제주대학교 공과대학 4호관 3층	http://localhost/ServiceProvider/Service3	제공
3	제주대학교 공과대학 4호관 4층	http://localhost/ServiceProvider/Service	제공

그림 66. 서비스 공급자 검색



서비스 공급자 검색의 성능분석은 그림 67과 같다. 성능분석은 서비스 요청자에서 서비스 저장소로 공급자 정보를 검색하고 응답받는데 소요되는 시간이다. 그래프의 가로축은 수행 횟수를 나타내고 세로축은 수행시간(초)을 나타낸다. 각각의 수행횟수 별로 0.069004, 0.0590034, 0.0750042, 0.0730042, 0.0730042, 0.0610035, 0.0720041, 0.0740042, 0.0730042, 0.0690039, 0.0740043, 0.0580033, 0.0730042, 0.0780044, 0.0770044, 0.0580033, 0.0590034, 0.0590033, 0.0750043, 0.0580033 초의 수행시간이 측정되었고 최소 수행시간은 0.0580033 초이고, 최대 수행시간은 0.0780044 이다. 평균적으로 서비스 요청자에서 서비스 공급자를 검색하는데 소요되는 시간은 0.068353905 초로 분석되었다.

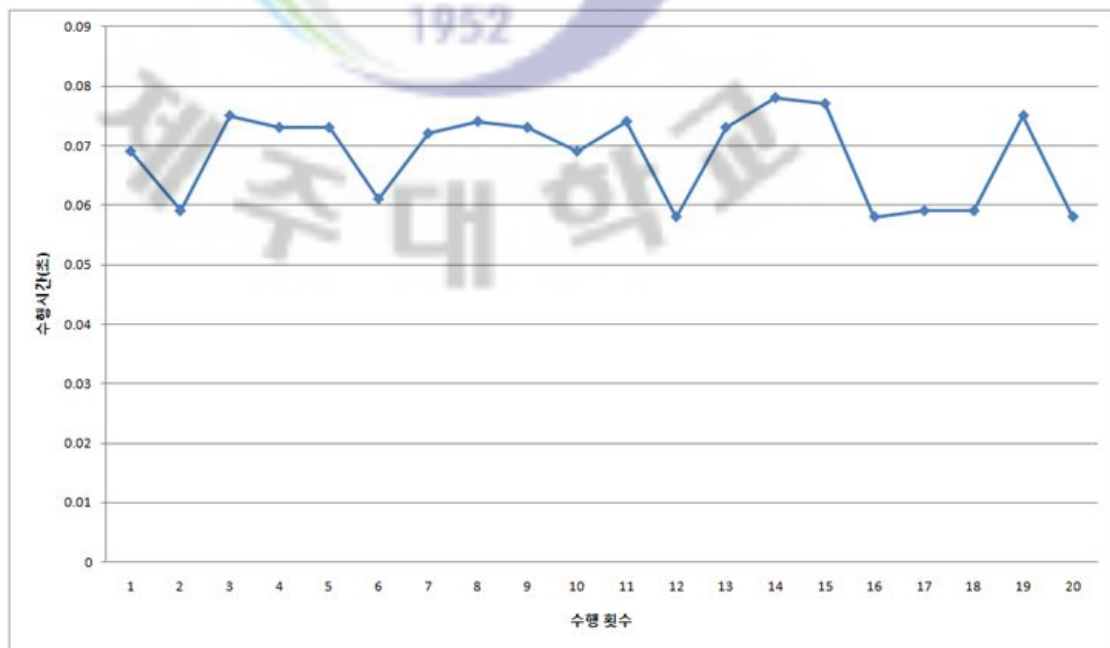


그림 67 서비스 공급자 검색의 소요시간

콘텐츠 요청자의 화면은 그림 68, 69, 70, 71과 같다. 콘텐츠 제공은 서비스 요청자에서 서비스 공급자에 접속하여 콘텐츠를 제공받는 단계이다. 실험은 제주대학교 공과대학 4호관의 1, 2, 3, 4층을 대상으로 하였다. 실험을 위해 서비스 공급자에 접속하여 지도정보 및 노드정보를 응답받고 총 20회에 걸쳐 콘텐츠를 제공 하였고 공

급자의 응답에 소요되는 시간을 측정하였다.

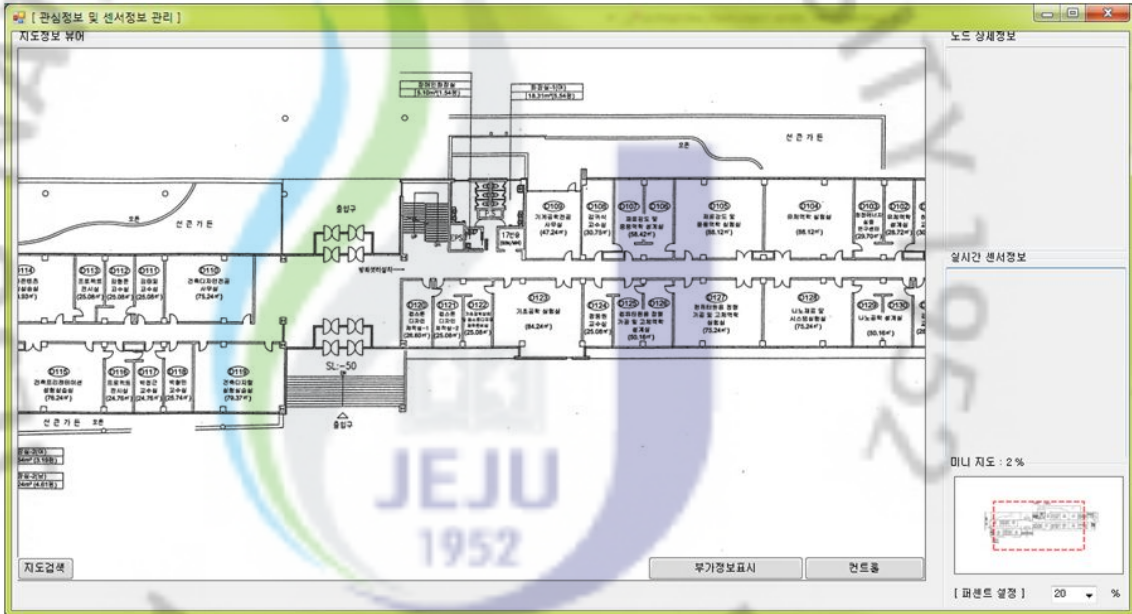


그림 68. 서비스 요청자 (공과대학 1층)

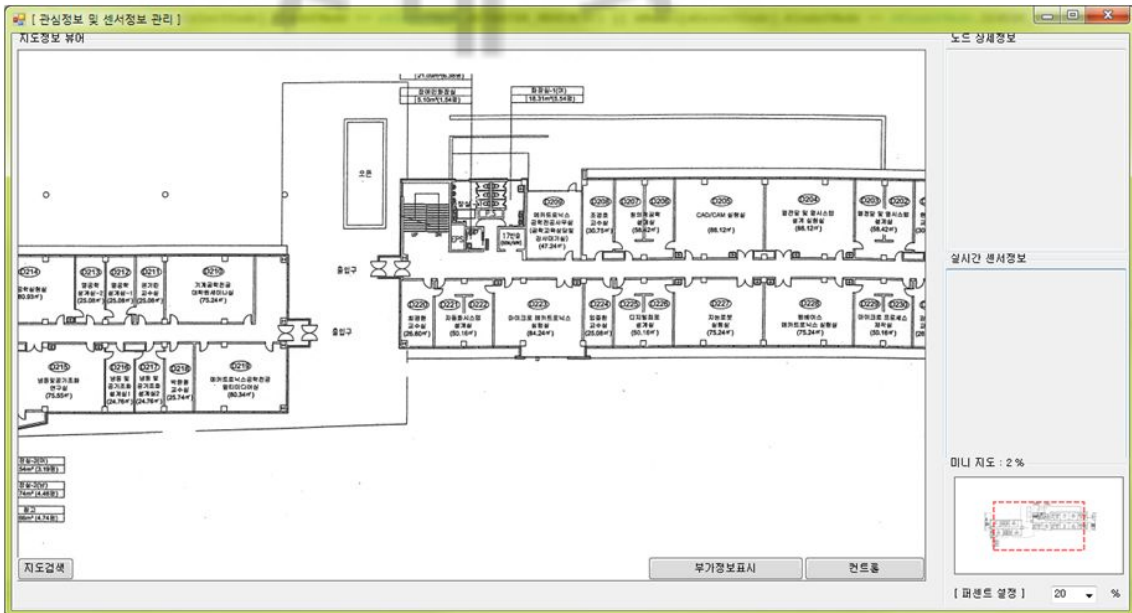


그림 69. 서비스 요청자 (공과대학 2층)

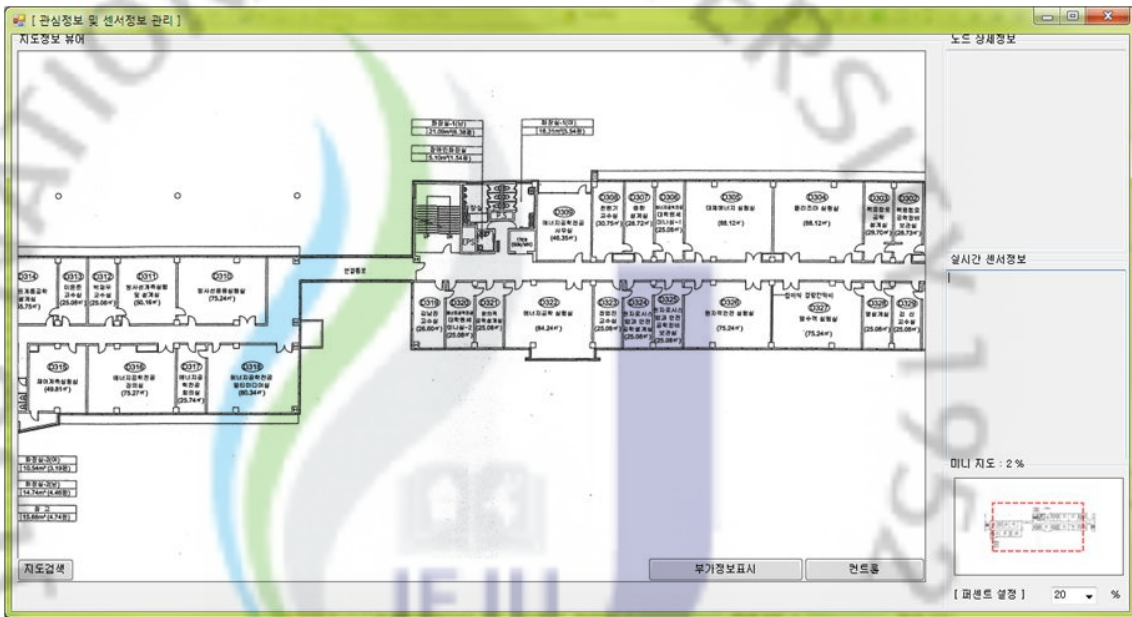


그림 70. 서비스 요청자 (공과대학 3층)

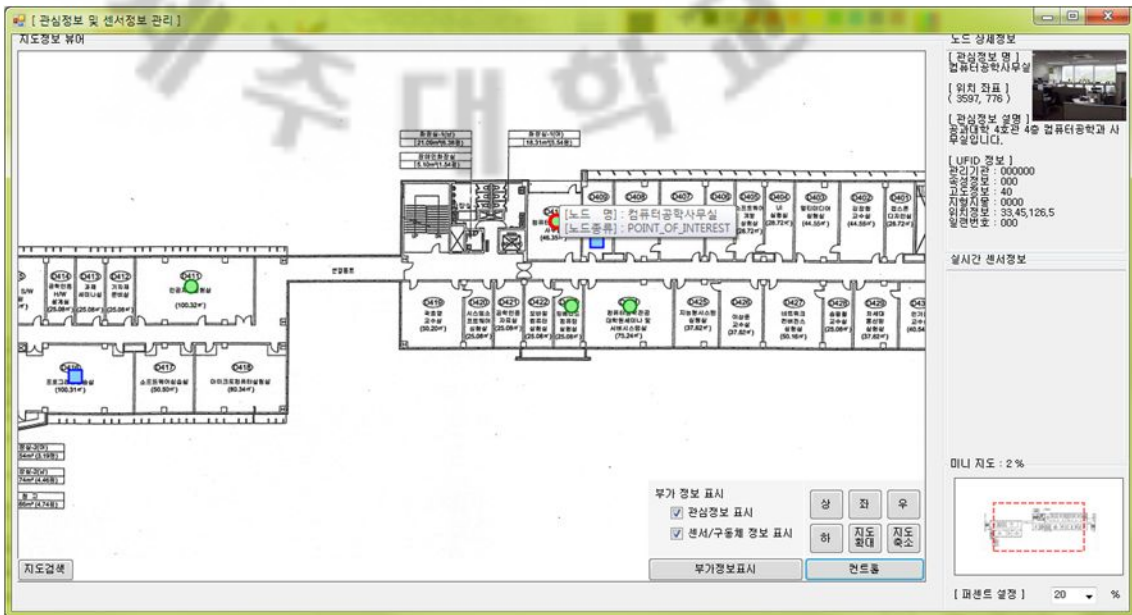


그림 71. 서비스 요청자 (공과대학 4층)

콘텐츠 제공의 성능분석은 그림 72와 같다. 성능분석은 서비스 공급자에서 서비스 요청자로 지도, POI, 센서, 구동체등의 콘텐츠 정보를 전달하는데 소요되는 시간이 다. 그래프의 가로축은 수행 횟수를 나타내고 세로축은 수행시간(초)을 나타낸다.

각각의 수행횟수 별로 0.744046, 0.7310441, 0.7260415, 0.734042, 0.7440425, 0.7210413, 0.7350443, 0.7350432, 0.7250415, 0.7250415, 0.7200411, 0.7400434, 0.7260415, 0.7310418, 0.7310418, 0.7200412, 0.7500429, 0.7320419, 0.7410424, 0.734042 초의 수행시간이 측정되었고 최소 수행시간은 0.7200411 초이고, 최대 수행시간은 0.7500429 이다. 평균적으로 서비스 요청자가 서비스 공급자에서 콘텐츠를 응답받는데 소요되는 시간은 0.732292395 초로 분석되었다.

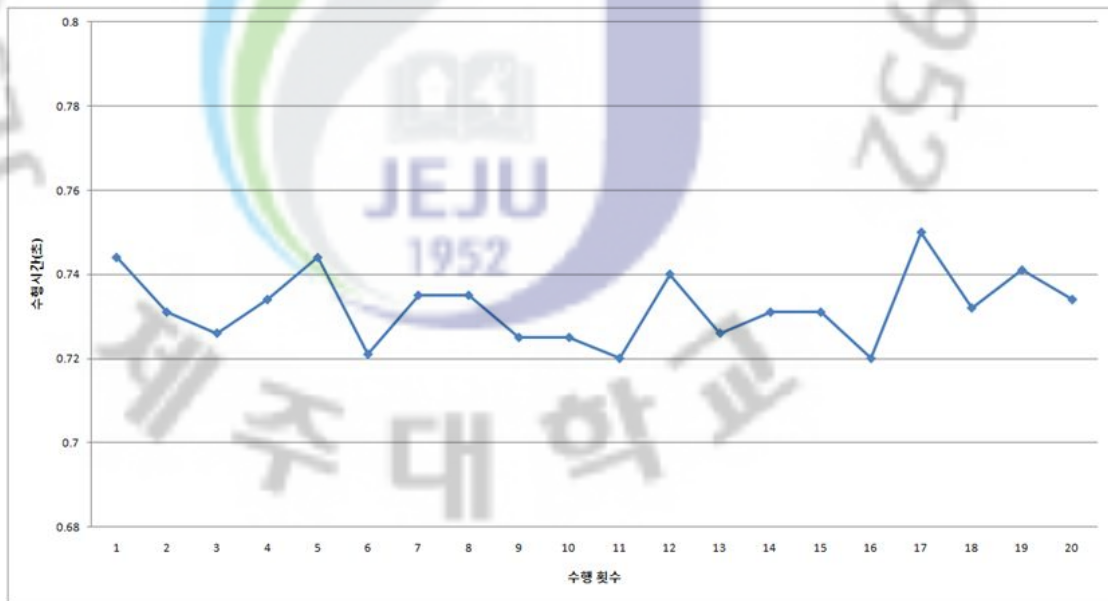


그림 72 콘텐츠 제공의 소요시간

## V. 결론

본 논문은 다양한 센서 네트워크로부터 실시간으로 수집되는 상황 데이터를 통합하고 처리한 정보를 제공하는 서비스 공급자를 위해 서비스 콘텐츠를 생성하고 관리하는 도구인 COST를 설계하고 구현한다.

기존의 SOA 기반의 센서웹에서 서비스 공급자는 매우 다양한 실시간 상황 정보를 제공하며 이를 관리하는 역할을 담당하며, 이를 위해서는 서비스 공급자별로 지도, POI(Point of Interest), 센서 노드, 구동체 노드 등을 포함하는 고유의 서비스 콘텐츠가 필요하며, 개별적으로 맞춤형 콘텐츠를 생성하고 유지관리하기 어렵다. 이를 위해 기존의 SOA 기반의 센서웹 기술을 고찰하였으며, 전송을 위한 기술 중 WCF를 채택하여 서비스 별로 OPEN API를 정의하였다. 실제로 콘텐츠 생성 및 관리를 위한 COST를 구현하고 기존 SOA 기반의 센서웹 시스템과 연동하고 콘텐츠 생성과 전송 모듈에 대해 성능분석을 하였다. 서비스 공급자 및 서비스 요청자를 위한 프로그램 배포기술로는 Click Once와 Smart Client를 적용해 보다 안정적이고 간편하게 프로그램 및 버전 관리의 편의성을 도모하였다.

본 논문에서 설계 및 구현한 COST는 SOA 기반의 센서웹에서 콘텐츠 생성을 분리하여 서비스 공급자의 일괄적인 관리가 용이하며, 전문적인 콘텐츠 생성이 가능하다. 서비스 공급자 입장에서는 콘텐츠 생성에 대한 부담이 줄어들어 서비스 제공에 전념할수 있으므로 보다 안정적으로 서비스 제공을 할 수 있는 환경이 제공된다. 또한 기존의 센서웹에 적용해 봄으로써, 향후 실시간 상황 정보를 수집하는 사용자에게 가시화하는 여러 시스템에도 적용 가능할 것으로 전망된다.



## 참고문헌

- [1] 이충호외 3인 "u-GIS 공간정보 기술 동향", 전자통신동향분석 제 22권 제 3호 2007년 6월
- [2] Mike Botts, "OGC® Sensor Web Enablement: Overview and High Level Architecture," Open GISConsortium, 2006.
- [3] Xingchen Chu and Rajkumar Buyya, "Service Oriented Sensor Web," In: Sensor Network and Configuration: Fundamentals, Standards, Platforms, and Applications, N.P. Mahalik (ed.), Springer-Verlag, Germany, 2007.
- [4] <http://www.opengeospatial.org/projects/groups/sensorweb>
- [5] 장윤섭외 2인 "지리공간 웹 기술 동향", 전자통신동향분석 제 22권 제 3호 2007년 6월
- [6] 이성규, 진찬욱, 김태석 ".NET Framework를 서비스 플랫폼으로 사용한 SOA 모델 구현 및 성능분석" 한국시물레이션학회 논문지 Vol. 16, No. 4, pp. 33-41 (2007. 12)
- [7] 이권민, 김정현, 박상준, 이성호, 신동욱, 구용완 "실시간 서비스 제공을 위한 SOA 기반의 웹서비스 설계"
- [8] <http://www.simpleisbest.net/archive/2007/12/03/1978.aspx> (유경상 WCF)
- [9] 이상훈, 한국건설기술연구원 "UFID 및 RFID 주소체계 소개 및 u-City 활용방안" u-City 포럼 기술분과 세미나 2005년 10월
- [10] 강민구 "전자태그(RFID)와 지형지물표시(UFID) 연구" 한국 인터넷 정보학회 (제4권 제3호)
- [11] "Smart Client Developer Center" Site, <http://www.microsoft.com/korea/msdn/smartclient>
- [12] "MSDN ClickOnce 배포전략" <http://msdn.microsoft.com/ko-kr/library/71baz9ah%28VS.80%29.aspx>