

# 전자해도 기반 Shape의 종류, 속성, 값의 조합을 이용한 설명 필드 추가 알고리즘 개발

## Developing an Algorithm that Adds a Description Field by Combining Type, Attribute, and Value Based on ENC

전혜연\* · 심문보\*\* · 강호윤\*\* · 유경완\*\* · 이현욱\*\*

Hae Yeon Jeon · Moon Bo Shim · Ho Yun Kang · Kyung Wan Yoo · Hyun Uk Lee

**요약** 본 연구의 목적은 전자해도(ENC)의 객체, 속성, 값의 조합에 따른 의미를 쉽게 파악하기 위한 알고리즘을 제시하고자 함이다. 최근 해양정보 분석을 위해 지리정보시스템(Geographical Information System, GIS)에서 전자해도에 대한 활용 요구가 증대되고 있다. 새로운 알고리즘 모색은 이러한 수요에 맞추기 위해 필수적인 작업이라고 할 수 있다. 본 연구에서 중점적으로 다루는 문제점은 전자해도를 Shape 포맷으로 변경할 경우 전자해도의 속성으로 입력된 S-57 Attribute ID가 의미적 설명 없이 그대로 변환된다는 점이다. 그로 인해 속성과 값의 조합으로 피처의 의미를 파악하는데 어려움이 많다. 본 연구에서는 이러한 문제를 해결하기 위해 새로운 알고리즘을 제시하고자 하였다.

**키워드** Shape 처리 알고리즘, Shape & 전자해도, S-57 Attribute “ID”를 “Meaning”으로 대체, 2.1.2 전자해도 속성 조합으로 설명필드 추가

**Abstract** The purpose of this study is to present an algorithm for easily identifying the meaning according to the combination of objects, attributes, values of the Electronic Navigational Chart(ENC). Recently, the need to utilize the ENC in the geographic information system (Geographic Information System, GIS) for marine information analysis has been increased. Exploring new algorithms is an essential task to meet this demand. The problem focusing on this study is that when changing the ENC to shape format, the S-57 Attribute ID input as the properties of the ENC is converted as is without semantic explanation. This makes it difficult to determine the meaning of a feature by a combination of attributes and values. In this study, we suggest descriptive algorithm to solve this problem.

**Key words**

## 1. 서론

전자해도(Electronic Navigational Chart, ENC)는 해안선, 등심선, 수심, 등대 및 항로표지 등 해도의 모든 정보를 국제수로기구(IHO) S-57 표준에 따라 개발한 디지털 해도이다. 현재는 차세대 전자해도 S-101 표준이 개발되었고 이를 기반 한 전자해도 또한 간행되었으나 아직까지는 사용이 국·내외적으로 일반화되지 않았다. 전자해도는 전자해도표시 시스템(Electronic Chart Display and Information System, ECDIS) 등에

탑재하여 항해목적으로 사용되기도 하지만 지리정보시스템(Geographical Information System, GIS)에서도 활용요구가 증가하고 있다. 그러나 해도 관련 전문가 또는 사용자 아닌 이상 전자해도에 포함된 객체 및 속성과 값에 대한 조합의 의미를 파악하는데 어려움이 많은 실정임에 따라, 본 연구에서는 전자해도의 객체, 속성, 값의 조합에 따른 의미를 쉽게 파악하기 위한 프로그래밍적인 방법을 제시한다.

## 2. 연구내용

본 연구에서는 ArcGIS Pro에서 전자해도를 Shape파일로 변환하였을 때 나타나는 문제점을 분석하였고 이에 대한 개선 알고리즘을 제시하였다. 프로그램 개발은 ArcGIS Pro API를 이용하였으며 PYTHON 개발언어를 사용하였다.

### 2.1 전자해도를 Shape 파일 분석

전자해도는 침선, 간출암 등 피처를 객체(Object)로, 특성은 속성과 값으로 정의한다. 즉 해도도식을 객체·속성·값의 조합으로 표현한다.

전자해도 제작 및 표시 등 전용 소프트웨어는 피처와 속성의 종류, 값, 형태(점·선·면)에 따라 Feature & Portrayal과 같은 환경파일을 참조하여 화면에 심벌로 표출하기 때문에 사용자가 각 피처에 대한 정보를 쉽게 이해 할 수 있으나 대부분 GIS 프로그램은 점, 선, 면으로 표현되기 때문에 전자해도 기반 Shape 파일을 활용하는데 다소 어려움이 있다.

즉 전자해도(S-57)의 각 객체의 표현은 장비(SW)에서 결정된다. 또한 전자해도 파일 자체는 수로정보를 S-57에 정의된 객체 ID와 속성의 ID가 저장된다.

ArcGIS Pro는 전자해도를 Shape로 변환할 경우 속성 명을 S-57 Attribute ID 대신 Acronym으로 변경되거나 속성 값은 ID를 그대로 사용한다. QGIS SW도 이와 같다.

이는 ID로 입력된 속성 값의 의미를 파악하는데 시간이 많이 소요되고 다양한 분야에서 필요한 값을 활용하는데 제약이 될 수 있다. 또한, 여러 개의 속성조합의 의미를 파악하는 것조차 관련 전문가가 아닌 이상 어려움이 있다.

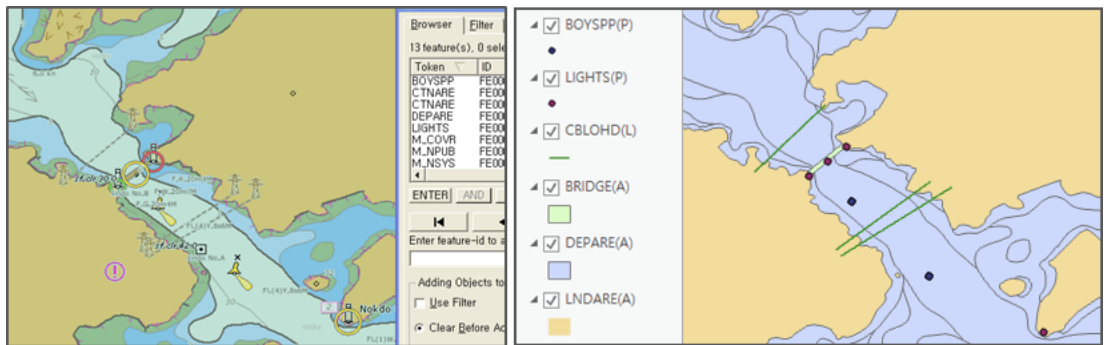
### 2.2 개선안 제시

이 같은 이유로 전자해도(S-57)기반 Shape 파일 종류 및 속성, 값, 형태를 분석하여 ID를 “Meaning”으로 대체하고 각 피처의 속성 조합이 의미하는 설명문을 추가하는 알고리즘을 제시한다.

#### 2.2.1 S-57 Attribute “ID”를 “Meaning”으로 대체 알고리즘

이 기능은 구현하기 위해 PYTHON의 Dictionary 자료형에 속성을 배열(LIST)로 추가하는 방식으로 S-57 Dictionary를 설계하였다.

Dictionary 자료형은 immutable한 키(key)와 mutable한 값(value)으로 맵핑되어 있는 집합체이며



전자해도 전용 소프트웨어

ArcGIS

Figure 1. 전자해도 전용 소프트웨어 & ArcGIS의 객체표현 비교

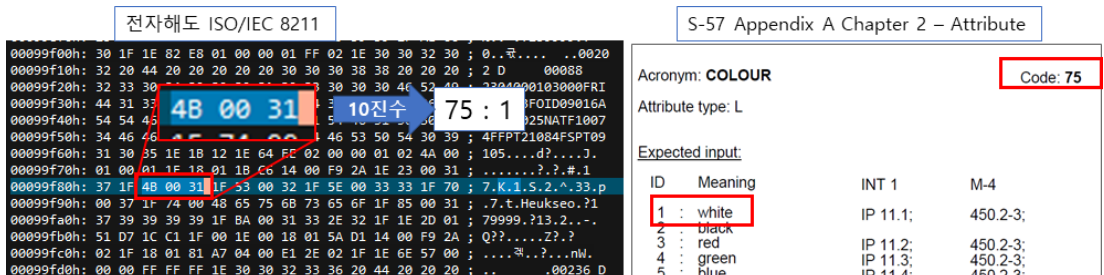


Figure 2. 전자해도의 속성 저장방식

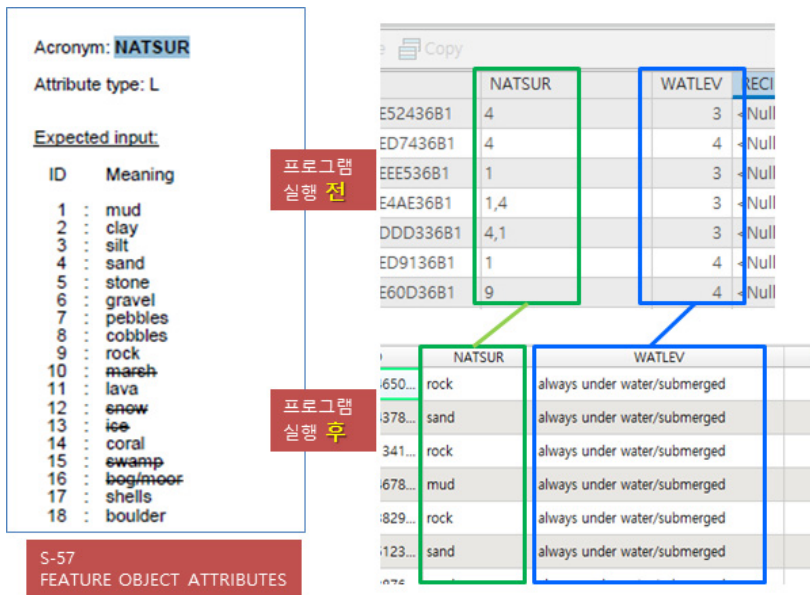


Figure 3. 프로그램 실행 전후 Shape 파일의 테이블 내용

보기1과 같이 ‘:’를 기준으로 왼쪽 “a”와 “b”는 Key의 역할이고 오른 쪽은 value로 구성된다.

```
보기 1 | Sample_dir = {"a" : 1, "b" : 2}
```

배열(LIST) 자료형은 여러 요소들을 갖는 집합으로 그 값을 자유롭게 변경할 수 있는 mutable한 데이터 타입이다. 배열은 첫 번째 요소부터 0,1,2,.. 순서로 할당되며, 배열의 값을 활용하기 위해서는 요소의 번호를 지정하여 사용한다.

```
보기 2 | Sample_list = ["SS", True, 21]
        ※ "SS"는 Sample_list[0] 에 입력되어 있다.
```

이 두 가지 자료형의 특성을 이용하여 Dictionary 자료형에 S-57 속성 Acronym을 Key로 “Meaning”을 value로 배열로 선언하여 S-57 속성에 대한 “구조체”를 정의하였다.

즉, 표와 같이 attList 이름의 Dictionary 타입을 선언하고 “CATOBS”, “COLOUR” 등과 같은 속성의 Acronym을 Key로 지정한다. mud, clay 등과 같은 “Meaning”을 배열로 정의한다.

다음 그림은 PYTHON에서 실제 적용한 사례이다. attList 변수를 Dictionary 자료형으로 선언하고 “CAT\_TS”, CATCBL“ 등 속성 Acronym을 Key으로 사용하였다. 또한 S-57 Appendix A Chapter 2 - Attributes를 참조하여 Key와 매칭 되는 값을 리스트

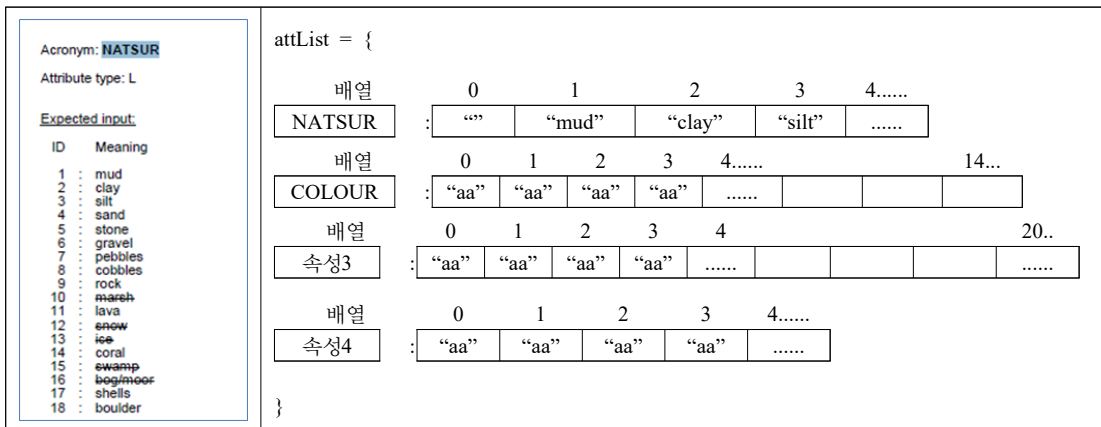


Figure 4. Dictionary 와 List를 이용한 구조체 정의

```

class s57Attri:
    def __init__(self):
        self.sublist = []
        self.attlist = {
            "CAT_TS": ["", "flood stream", "ebb stream", "Other tidal flow"],
            "CATCBL": ["", "power line", "telephone/telegraph", "transmission line", "telephone", "telegraph", "mooring cable/chain"],
            "CATCOA": ["", "steep coast", "flat coast", "sandy shore", "stony shore", "shingly shore", "glacier (seaward end)", "mangrove", "marshy shore",
            "CATFIF": ["", "fishing stake", "fish trap", "fish weir", "tunny net"],
            "CATPIP": ["", "pipeline in general", "outfall pipe", "intake pipe", "sewer", "bubbler system", "supply pipe"],
            "CATPRA": ["", "quarry", "mine", "stockpile", "power station area", "refinery area", "timber yard", "factory area", "tank farm", "wind farm",
            "CATSLC": ["", "breakwater", "groyne (groyne)", "mole", "pier (jetty)", "promenade pier", "wharf (quay)", "training wall", "rip rap", "revetment",
            "CATWAT": ["", "breakers", "eddies", "overfalls", "tide rips", "bombora"],
            "CATWRK": ["", "non-dangerous wreck", "dangerous wreck", "distributed remains of wreck", "wreck showing mast/masts", "wreck showing any portion",
            "COLOUR": ["", "white", "black", "red", "green", "blue", "yellow", "grey", "brown", "amber", "violet", "orange", "magenta", "pink"],
            "EXPSOU": ["", "within the range of depth of the surrounding depth area", "shoaler than the range of depth of the surrounding depth area", "deeper",
            "NATQUA": ["", "fine", "medium", "coarse", "broken", "sticky", "soft", "stiff", "volcanic", "calcareous", "hard"],
            "NATSUR": ["", "mud", "clay", "silt", "sand", "stone", "gravel", "cobble", "rock", "marsh", "lava", "snow", "ice", "coral", "swamp",
            "PRODCT": ["", "oil", "gas", "water", "stone", "coal", "ore", "chemicals", "drinking water", "milk", "bauxite", "coke", "iron ingots", "salt",
            "QUASOU": ["", "depth known", "depth unknown", "doubtful sounding", "unreliable sounding", "no bottom found at value shown", "least depth known",
            "RESTRN": ["", "anchoring prohibited", "anchoring restricted", "fishing prohibited", "fishing restricted", "trawling prohibited", "trawling rest",
            "STATUS": ["", "permanent", "occasional", "recommended", "not in use", "periodic/intermittent", "reserved", "temporary", "private", "mandatory",
            "WATLEV": ["", "partly submerged at high water", "always dry", "always under water/submerged", "covers and uncovers", "awash", "subject to inun
        }

```

Figure 5. S-57 Dictionary 구현 사례

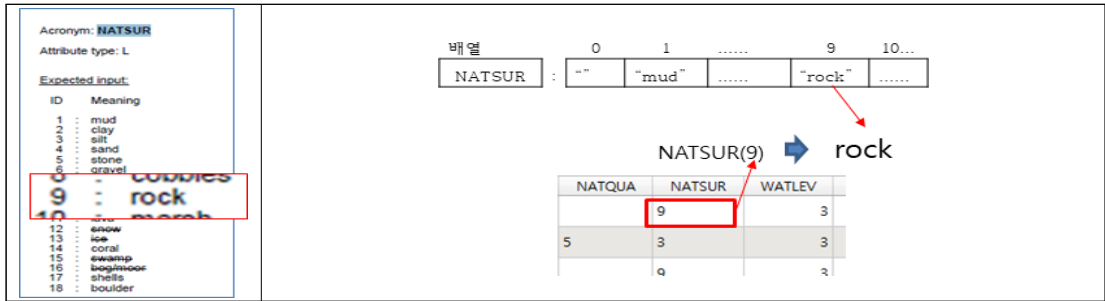


Figure 6. S-57 Attribute "ID"를 "Meaning"으로 대체 로직

자료형으로 정의하였다.

전자해도 기반 Shape의 필드 값으로 입력된 ID를 속성 값이 입력된 배열의 위치 값으로 활용하기 위해

리스트(배열)의 첫 번째 요소 즉 0번째는 "", 두 번째 위치부터 mud, clay... 값을 순차적으로 입력한다.

Shape의 필드 값으로 입력된 ID가 배열의 위치이기

```

def strToList(strtmp):
    string = str(strtmp)
    string.strip()
    start = 0
    item = []

    while(True):
        pos = string.find(",",start)
        # print("1 start = {0}, pos = {1}".format(start, pos))

        if pos > -1:
            tmp = string[start:pos]
            item.append(tmp)
            start += (pos-start) + 1
            # print("2 start = {0}, pos = {1}".format(start, pos))

        else:
            tmp = string[string.rfind(",") + 1 : ]
            tmp.strip()
            item.append(tmp)
            # print("The end")
            return item

```

Figure 7. 연속된 문자열을 정수형 배열로 변경하는 로직

```
with arcpy.da.UpdateCursor(outFeatureClass, [att_name, en_field_name]) as cursor:
    for row in cursor:
        aa = str(row[0])
        if aa == ' ':
            continue

        att_li = strToList(aa)
        row[1] = self.chang_AttriInfo(att_li, att_name)
        cursor.updateRow(row)
```

Figure 8. S-57 “Meaning” 값으로 변경하는 알고리즘

BCNSHP	CATCAM	COLOUR	COLPAT
5	4	6,2,6	1
5	4	6,2,6	1
5	3	6,2	1

→

BCNSHP	CATCAM	COLOUR	COLPAT
pile beacon	east cardinal mark	black,yellow,black	horizontal bands from...
pile beacon	east cardinal mark	black,yellow,black	horizontal bands from...
pile beacon	south cardinal mark	yellow,black	horizontal bands from...

프로그램 실행 전

프로그램 실행 후

Figure 9. 프로그램 실행 전·후

때문에 정수형이어야 한다. 이에, 입력되는 필드 값의 자료형을 검사하여 “1,2,1”과 같이 연속적인 문자형인 경우는 모두 독립된 숫자로 변형시켜야 하다.

다음 그림은 PYTHION에서 실제 적용한 사례이다. strToList() 함수는 “1,2,1”과 같은 문자를 “,”를 기준으로 각각 분리시켜 배열[“1”,“2”,“3”]에 저장시켜 주는 로직이다.

다음 알고리즘은 Shape 테이블을 Record 단위로 Loading하고 필드에 입력된 숫자를 S-57 “Meaning” 값으로 변경하는 알고리즘이다. 필드 값이 “1,2,1”과 같은 문자일 경우 strToList()함수를 호출하여

[“1”,“2”,“1”]과 같은 배열로 변환을 한 후 chang\_AttriInfo() 함수를 호출하여 S-57 “Meaning” 값으로 변경한다. 만약 “COLOUR”가 “1,2,1”로 입력되었다면, “white, black, white”로 변경되어 Shape 테이블을 업데이트 한다.

아래 그림은 제시된 알고리즘을 이용해 처리한 결과의 전·후 비교이다.

### 2.2.2 전자해도 속성 조합으로 설명필드 추가

중이해도는 장애물, 침선, 해안선 등을 국제기준(IHO S-4, INT1)에 따라 도식으로 표현하지만 전자해

**해도도식, INT1**

- 암암  
수심이 알려진 위험한 암암  
Dangerous underwater rock of known depth

- 암암  
기본수준면에서 보이지 않고 수심을 알 수 없는 위험한 바위  
Dangerous underwater rock of uncertain depth

**S-57**

객체(레이어)	속성(field)	값
UWTROC	VALSOU	0.9
	WATLEV	3
	NATSUR	9
	QUASOU	1

객체(레이어)	속성(field)	값
UWTROC	VALSOU	3
	WATLEV	9
	NATSUR	3
	QUASOU	2

QUASOU	WATLEV	VALSOU	K_Inform	E_Inform
depth unknown	always under water/...	0	암암(기본수준면에서 보이지 않고 수심을 알 수 없는 위험한...	Dangerous underwater rock of uncertain depth
depth unknown	always under water/...	0	암암(기본수준면에서 보이지 않고 수심을 알 수 없는 위험한...	Dangerous underwater rock of uncertain depth
depth known	always under water/...	0.9	수심이 알려진 위험한 암암	Dangerous underwater rock of known depth
depth known	always under water/...	8.6	수심이 알려진 위험한 암암	Dangerous underwater rock of known depth

Figure 10. IHO S-57 및 해도도식을 기반으로 한 설명 필드

CATOBS	COND1	EXPSOU	QUASOU	VALSOU	K Inform	E Inform
fish haven		shoaler than range o...	least depth known	8	측심에 의해 최소수심이 알려진 여초구역	Fish haven, , least de...
fish haven		shoaler than range o...	least depth known	4.8	측심에 의해 최소수심이 알려진 여초구역	Fish haven, , least de...
fish haven		within the range of...	least depth known	12.2	측심에 의해 최소수심이 알려진 여초구역	Fish haven, , least de...
fish haven		shoaler than range o...	least depth known	8.2	측심에 의해 최소수심이 알려진 여초구역	Fish haven, , least de...
			depth unknown	0	장애물	Obstructions
		shoaler than range o...	least depth known	4.8	주변 수심 범위 밖의 얇은 장애물	Obstructions, thin...

Figure 11. 설명필드(K\_Inform, E\_Inform) 예시

도는 객체 및 속성으로 표현한다. 이에, 전자해도의 객체 및 속성을 그대로 포함한 Shape의 레이어명과 필드명, 속성 값을 이용하여 IHO S-57 표준과 국내 해도도식에 기반으로 한 설명내용을 입력했다.

Shape 파일은 전자해도의 객체 및 속성을 그대로 포함하도록 제작되었기 때문에 S-57 표준에 익숙하지 않은 사용자는 속성의 의미파악이 어려울 수 있다. 이에, 속성 조합 결과를 K\_Inform(한글), E\_Inform(영문) 필드에 추가하였다.

다음 표는 국내외 표준을 근거로 레이어 및 필드명과 속성 값에 따라 적용된 설명필드(K\_Inform(한글), E\_Inform(영문))이다. 즉, 간출암은 UWTRC 객체(레이어)와 속성 값이 WATLEV = 4 and NATSUR=9 으로 정의된다.

위 표를 기반으로 Shape 테이블에서 데이터를 레코

드 단위로 읽어들어 속성(필드)값의 조합에 따라 설명 필드를 생성하여 테이블에 적용하기 위한 로직은 다음과 같다.

- ① Shape 테이블에 속성조합에 의해 생성된 값을 입력할 필드를 생성한다. 한글은 K\_inform, 영문은 E\_inform으로 정의했다.
- ② Shape의 명칭(종류)이 “OBSTRN”이면 ③ 방향으로 처리되고, 아니면 다음 객체인지를 비교한다. 이와 같은 프로세스를 통해 해당되는 객체의 속성 조합으로 INT1에서 정의한 설명 값을 구할 수 있다.
- ③ 조합에 필요한 속성들을 Shape 테이블에서 불러오기 위해 배열로 선한다.
- ④ Shape 테이블에서 필요한 속성들을 읽어온다.
- ⑤ GetOBSTRN() 함수를 호출하여 입력받은 속성

Table 1. 객체 및 필드 값 조합에 따른 설명 예

식	항목도	설명 필드 K_Inform, E_Inform	객체 Layer	필드:값	
	간출암 Rock which covers and uncovers		UWTRC	WATLEV NATSUR	4 9
	기본수준면에서 씻기는 바위 Rock awash at the level of chart datum		UWTRC	WATLEV NATSUR VALSOU	5 9 0
	수심이 알려진 위험한 암암 Dangerous underwater rock of known depth		UWTRC	NATSUR WATLEV QUASOU	9 3 1
	항해에 위험하지 않은 수심이 알려진 바위 Underwater rock of known depth, not dangerous to surface navigation		UWTRC	NATSUR WATLEV VALSOU	9 3 is not null
	암암(기본수준면에서 보이지 않고 수심을 알 수 없는 위험한 바위) Dangerous underwater rock of uncertain depth		UWTRC	WATLEV NATSUR QUASOU	3 9 2
	주변 수심 범위내 존재 하는 암초 within the range of depth of the surrounding depth area		UWTRC	NATSUR WATLEV QUASOU EXPSOU	9 3 1 1
	주변 수심보다 얇은 암초 shoaler than the range of depth of the surrounding depth area		UWTRC	NATSUR WATLEV QUASOU EXPSOU	9 3 1 2

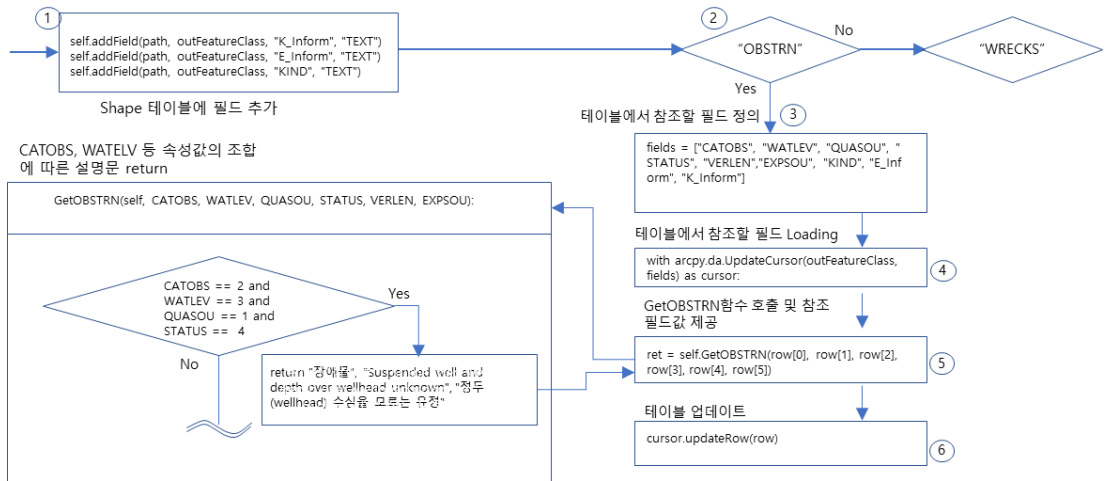


Figure 12. 객체의 속성 조합에 따른 INT1 설명문을 추가하는 로직

값의 조합들을 분석한 후 해당되는 값을 return 한다.

- ⑥ return된 결과 값을 Shape 테이블에 입력하기 위해 cursor.updateRow(row)를 호출한다.

### 3. 결론

최근, 육상의 지리정보와 함께 해양의 지리정보 또한 다양한 분야에서 활용 요구가 증가하고 있는 현황이다. 이에 본 논문에서는 전자해도를 지리정보에서 많이 사용되는 Shape 형태로 변환하였다. 전자해도 기반 Shape파일은 “CALPAT : 3” 과 같이 전자해도의 속성과 ID값이 그대로 입력됨에 따라 의미를 파악하는데 불편과 시간이 많이 소요되고, 여러 개의 속성조합이 의미를 파악하는 것조차 관련 전문가가 아닌 이상 어려움이 있다. 따라서 ID를 ID가 가리키는 문자로 입력하고 속성조합에 따른 설명을 추가함으로써 전자해도 기반 Shape를 활용하는 데 도움이 될 것으로 기대한다.

### 참고문헌

1. IHO Transfer Standard for Digital Hydrographic Data (Edition 3.1.0, November 2000) - Main Document
2. IHO S-57 Edition 3.1 Appendix A, Chapter 1 - Object Catalogue
3. IHO S-57 Edition 3.1 Appendix A, Chapter 2 - Attribute Catalogue
4. IHO INT 1 Symbols and Abbreviations used on Paper Charts (current English version, Edition 8, 2020)
5. 국립해양조사원 해도도식 2015년 12월 개정판

