

## 개요

고장수목분석을 수행할 때 생성되는 고장수목을 효과적으로 분석 및 저장하기 위한 XML 스키마의 정의를 소개한다. 이는 일반적인 고장수목과 이를 확장한 경우도 잘 나타내고 저장할 수 있다. 또한 제안한 XML 스키마를 이용하여 Function Block Diagram의 고장수목을 저장한 사례를 보인다.

## 목적

- FT의 저장에 적합한 XML 스키마의 정의
- 다양한 형태의 FT의 저장

## 배경

### · FBD

PLC(Programmable Logic Controller)용 표준 프로그램 언어 중 하나로 PLC에서 동작하는 소프트웨어를 개발하는데 사용되는 언어다. 미리 정의된 블록들의 연결로 동작 방식을 표현한다.

### · FTA

고장수목분석(FTA, Fault Tree Analysis)은 시스템의 위험성을 분석하는데 사용되는 기법으로 어떤 특정한 예상 사고에 대하여 그 사고의 원인이 되는 장치/기기의 결함이나 오류를 순차적, 도식적으로 검토 및 분석하여 시스템의 위험성을 평가하는 방법이다. 이 기법은 트리를 이용해서 시각적으로 분석 결과를 보여주며, 최상위 이벤트인 시스템의 고장으로부터 최하위 이벤트인 고장의 원인을 역으로 찾아가는 방식으로 시스템의 위험을 분석한다.

## 스키마 정의

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://dsLab.konkuk.ac.kr/FaultTreeSchema/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://dsLab.konkuk.ac.kr/FaultTreeSchema/">

  <xs:element name="root" type="tns:rootType"/></xs:element>

  <xs:element name="node" type="tns:nodeType"/></xs:element>

  <xs:element name="faultTree" type="tns:faultTreeType"/></xs:element>

  <xs:complexType name="rootType">
    <xs:sequence maxOccurs="unbounded" minOccurs="0">
      <xs:element name="node" type="tns:nodeType"/></xs:element>
    </xs:sequence>
    <xs:attribute name="data" type="string"/></xs:attribute>
    <xs:attribute name="desc" type="string"/></xs:attribute>
    <xs:attribute name="id" type="string"/></xs:attribute>
    <xs:attribute name="type" type="string"/></xs:attribute>
  </xs:complexType>

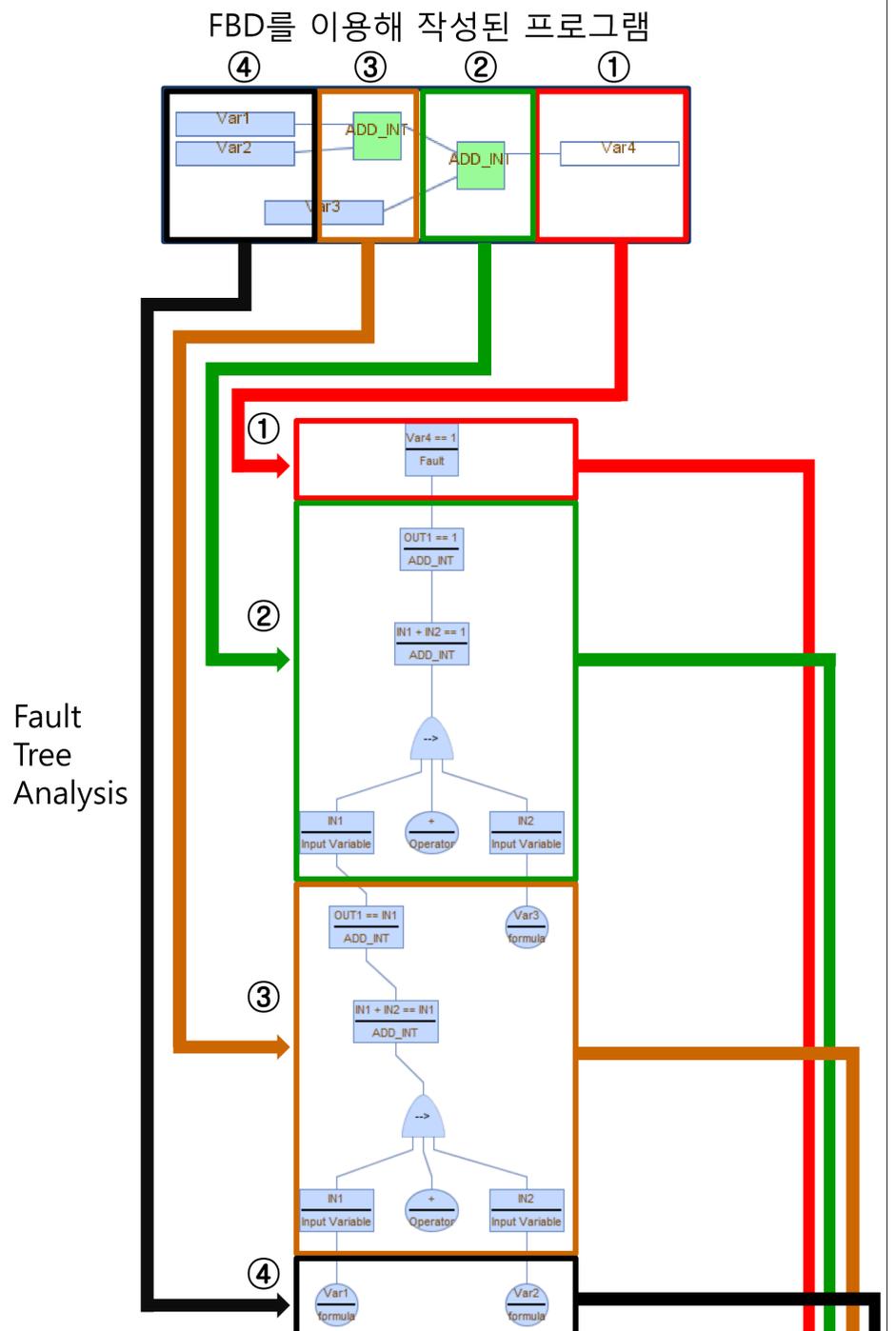
  <xs:complexType name="nodeType">
    <xs:sequence maxOccurs="unbounded" minOccurs="0">
      <xs:element name="node" type="tns:nodeType"/></xs:element>
    </xs:sequence>
    <xs:attribute name="data" type="string"/></xs:attribute>
    <xs:attribute name="desc" type="string"/></xs:attribute>
    <xs:attribute name="id" type="string"/></xs:attribute>
    <xs:attribute name="type" type="string"/></xs:attribute>
  </xs:complexType>

  <xs:complexType name="faultTreeType">
    <xs:sequence>
      <xs:element name="root" type="tns:rootType"/></xs:element>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

<b>data</b>	FT에서 각 노드가 가지는 공식 또는 서술적인 형태의 실제 정보
<b>desc</b>	data 만으로 표현하기 어려운 경우를 위한 정보에 대한 부가 설명
<b>id</b>	각 노드들의 순서 및 구분을 위한 id
<b>type</b>	게이트나 단말 노드 등 노드 종류의 구분을 위한 값

## 결과



FTA를 통해 생성된 FT

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<tns:faultTree xmlns:tns="http://dsLab.konkuk.ac.kr/FaultTreeSchema/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://dsLab.konkuk.ac.kr/FaultTreeSchema/ ../com.DSLab.KU.FBDFTA/FaultTreeSchema.xsd">
  <root data="Var4 == 1" desc="Fault" id="1" type="EVENT_BASIC">
    <node data="OUT1 == 1" desc="ADD_INT" id="2" type="EVENT_INTERMEDIATE">
      <node data="IN1 + IN2 == 1" desc="ADD_INT" id="3">
        <node data="" desc="ADD_INT" id="4" type="GATE_ORDER">
          <node data="IN1" desc="Input Variable" id="5" type="EVENT_INTERMEDIATE">
            <node data="OUT1 == IN1" desc="ADD_INT" id="9">
              <node data="IN1 + IN2 == IN1" desc="ADD_INT" id="10">
                <node data="" desc="ADD_INT" id="11" type="GATE_ORDER">
                  <node data="IN1" desc="Input Variable" id="12" type="EVENT_INTERMEDIATE">
                    <node data="Var1" desc="formula" id="15" type="EVENT_BASIC" />
                  </node>
                  <node data="" desc="Operator" id="13" type="EVENT_BASIC" />
                  <node data="IN2" desc="Input Variable" id="14" type="EVENT_INTERMEDIATE">
                    <node data="Var2" desc="formula" id="16" type="EVENT_BASIC" />
                  </node>
                </node>
              </node>
            </node>
          </node>
        </node>
      </node>
    </node>
  </root>
</tns:faultTree>
```

스키마를 따라 저장된 FTA

## 결론 및 향후 연구

본 논문에서 제안한 방식은 기존의 FTA에 사용되는 도구에서도 FT를 저장하는데 사용될 수 있을 것이다. 이는 FT의 생성 과정이나 분석 결과를 더욱 이해하기 쉽게 도와줄 것이며, XML을 이용함으로써 이 결과를 다른 시스템에서도 쉽게 사용할 수 있을 것이다.