

OOP를 이용한 배전자동화복구시스템의 설계

김필석 이승재
*명지대학교

조정현 박종국 안복신
**LG산전

OOP-Based Design of Service Restoration System in Distribution Automation System

*Pil-Suk Kim Seung-Jae Lee
*Myong-Ji Univ.

Jung-Hyun Cho Jong-Kuk Park Bok-Sh
**LG Industrial Systems Lab.

Abstract - This paper proposes the OOP design for development of the service restoration system in DAS. It utilizes the Booch Method and has features of high reusability, expansion and manatenance capability.

1. 서 론

최근 그 적용이 확산되고 있는 Object-Oriented Programming (OOP) 은 기존 software 개발방법과는 달리 Object의 개념을 중심으로 각 Object들간의 상호작용을 program 하는 것으로서 data와 operation간의 상호중속성을 최소화함으로써 code의 재사용성을 높이고 시스템의 유지, 보수 및 확장을 용이하게 할 수 있다는 장점을 갖고 있어 대규모 시스템개발에 많이 이용되고 있으며 전력계통의 여러 응용소프트웨어 개발에도 그 적용이 활발히 이루어지고있다 [1,2]. 배전자동화시스템은 현재 사고 복구등 일부 응용프로그램의 개발적용이 되고는 있으나 소프트웨어의 측면에서는 아직 초기단계로서 앞으로 진단, 부하균등화, 과부하해소 등 다양한 응용소프트웨어의 개발 및 통합운용 환경의 개발이 요구되며 이를 위하여 배전자동화 프로젝트 전체적으로 본 효율적인 개발방안이 필요하다.

본 논문에서는 [3]에서 제안된 피더 사고복구 알고리즘에 근거한 사고복구시스템을 구현하는데 있어서 OOP를 적용한 design 방향을 Booch Method [4] 에 근거하여 설계를 하였으며 이를 Booch Notation 을 이용하여 제시한다.

2. 본 론

[3] 에서 제시된 복구방안은 정전구역의 분할수 및 건전부하 절체여부에 따라 SGR (Single Group Restoration), DGR (Double Group Restoration), TGR (Triple Group Restoration), SGR_LT (SGR

with Load Transfer), DGR_LT (DGR with Load Transfer) 등 5가지 기본방안을 통하여 다수의 복구해를 구하고 이들의 Fuzzy 평가를 통하여 가장 좋은 해를 찾는 방안이다. 이와같은 사고복구 시스템의 개발을 위하여는 계통탐색기능이 필수적이며 이는 사고복구만이 아닌 다른 응용프로그램에서도 공통적으로 요구되는 기능이며 따라서 시스템개발 시 타 소프트웨어에서의 사용이 용이하도록 하여야 한다. 사고복구에서 다루는 피더는 스위치와 이로부터 분리되는 피더구간을 나타내는 zone 으로 구성되어 있어 이러한 실제적인 구조를 프로그램에서 쉽게 인식할 수 있도록 함이 시스템의 확장 및 변경에 용이하다. 또한 사고복구방안 및 Fuzzy 평가방안은 계통의 특성에 따라 달라질 수 있으므로 이의 수용을 위하여 이들이 프로그램의 다른 부분으로부터 분리가 되어있는 것이 바람직하다. 이러한 요구사항들을 고려하여 본 연구에서는 다음과 같은 OOP 설계안을 제시한다.

2.1 Class 범주

사고복구시스템에서 사용되는 class들을 기능상 크게 다음의 4가지로 구분하였다.

- a) System Data: 계통의 실제 구성요소 정보가 여기에 포함되며 사고복구계획 수립시에 이들의 능동적기능은 요구되지 않으므로 수동적인 data 로 규정한다.
- b) Network Search : 여러 구성요소들이 모여서 형성된 계통과 이들의 탐색기능을 갖는 Active 객체로 구성된다.
- c) Restoration Scheme : 복구계획을 구하는 여러 방안 및 그 방안의 절차가 정의된다.
- d) Fuzzy Evaluation: 구해진 복구해의 종합적 선호도를 다양한 operation - fuzzy membership degree구하기, fuzzy inference, defuzzification 등을 통하여 평가한다.

2.1.1 System Data

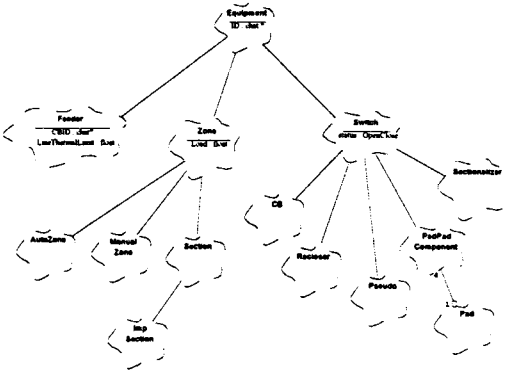


그림 1 Data 관련 Class Diagram

Equipment는 Feeder, Zone, Switch 등 3개의 하위 클래스와 Inheritance 관계를 갖는 상위클래스이다 (그림1). Switch는 on, off 상태정보를 가지고 있고 Zone은 switch로 둘러싸인 선로구간의 부하정보를 지닌 것으로 자동, 수동등 복구단계에 따라 요구되는 AutoZone, Manual Zone 과 Load Flow 계산시에 필요한 Section 의 Topology 정보를 담고 있다. Section의 하위 클래스인 ImpSection 은 Topology 정보와의 분리되어 Impedance 정보가 추가된 derived class 로서 프로그래밍상의 편의성을 위하여 도입되었다. Feeder는 각 급전선의 CB정보와 용량에 대한 정보를 제공한다.

2.1.2 Network Search

배전계통은 Vertex와 Link로 표현되는 일반적인 network으로 표현이 가능하므로 2개의 template을 가지는 parameterized class 로서 Network 이름을 갖는 최상위 class 를 정의하였다 (그림2). Network 클래스에는 일반적인 계통탐색 behavior를 method 로 제공되도록 하였으며 하위클래스들은 이를 상속 받아 parameter 의 대입을 통하여 (점선화살표) 목적에 맞는 탐색operation을 이용한다.

SectionNetwork와 ZoneNetwork의 차이점은 Vertex와 Link의 정의를 서로 반대로 하고 있다는 것으로서 SectionNetwork는 분기점을 취급하여 Vertex가 Switch가 되는 반면 ZoneNetwork는 Link를 Switch로 취급한다. 두 class는 서로 Vertex와 Link가 달라서 상위 Network class의 method를 호출하는 방법이 반대가 된다. 복구 scheme이 관심으로 하는 switch의 종류에 따라 ZoneNetwork은 2개의 instantiated class를 가지게 되며 SectionNetwork class는 1개의 instantiated class를 가지게 된다. Topology에 대한 method 를 SectionNetwork에서 상속받는 LoadFlowNetwork은 배전계통에서의 Load Flow 기능을 method로 제공

하고 있으며 Link로서 ImpSection을 취하는 DASFlowNetwork을 Instantiated class로 가진다.

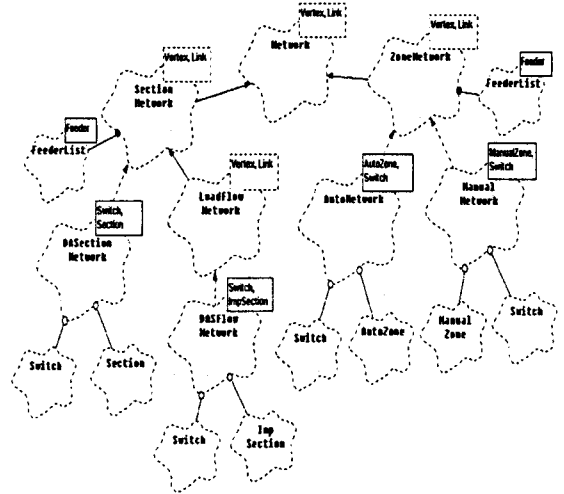


그림 2 Network 관련 Class Diagram

2.1.3 Restoration Scheme

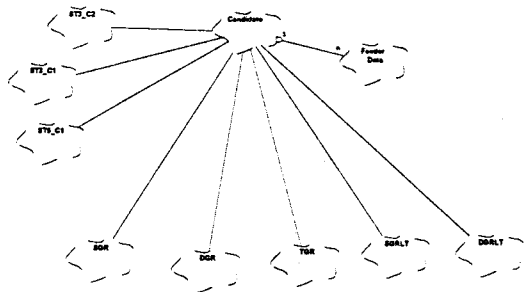


그림 3 Candidate Class Diagram

본 논문에서 다루고있는 5개의 복구기본방안은 switching 을 구하는 절차와 Backup Feeder수가 다를 뿐 그 기본절차는 동일하다. 따라서 최상위 클래스인 Candidate 에 공통적인 사항을 정의하고 switching을 구하는 method는 각 하위 class에서 정의되도록 한다(그림 3). Candidate class에 정의되는 공통사항에는 복구해평가를 위한 data를 구하는 method들과 이를 평가하기 위해서 Fuzzy Evaluation 관련 객체들을 using하는 method들이 정의되어 있다. Candidate class의 instance들의 state diagram은 그림 4와 같다. 그림 5는 SGR이 복구 switching을 구하는 세부과정을 나타내고 있는 Object Diagram 이다. SGR 객체의 FindSwitchingSequence()가 호출되면 계통을 나타내는 객체인 RestAutoNetwork에 backup feeder의 tie switch를 close 시키고 해당 backup feeder의 usefactor를 계산하는 RestAutoNetwork의 method 를 호출한다. 계산된 usefactor가 limit를 넘지 않을

