2005년도 대한전기학회 전력기술부문회 추계학술대회 논문집(2005.11.18~19)

직관적인 활동 다이어그램을 이용한 전력계통 운영의 자동화

신만철*, 오성균*, 황인준*, 김건중**, 전동훈**
(주)파워이십일*, 충남대학교**

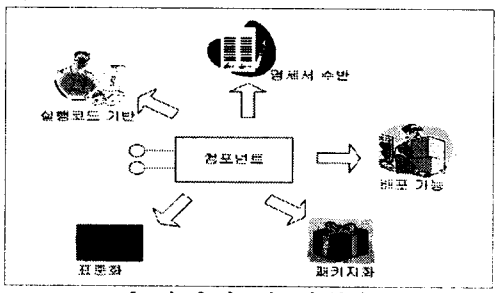
Automated operation of power system using intuitive activity diagram

M.C. Shin*, S.K. Oh*, I.J. Whang*, K.J. Kim**, D.H. Jeon
Power21 corp. ChungNam Nat' univ.

Abstract - 본 논문에서는 활동 다이어그램을 이용하여 전력계통 운영의 자동화에 대한 연구 수행 내용을 다루고 있다. 자동화를 가능케 하는 도구인 Smartflow의 기본 개념을 발전시키고 설계를 거쳐 사례연구를 수행한 결과 충분한 가능성을 확인할 수 있었으며, 현존하는 상용 프로그램과의 비교를 통해 비교 우위에 있다는 것도 알게 되었다. 전력계통에서의 활용 뿐 만 아니라 다른 시스템에도 적용이 가능함을 알 수 있었다.

1. 서 론

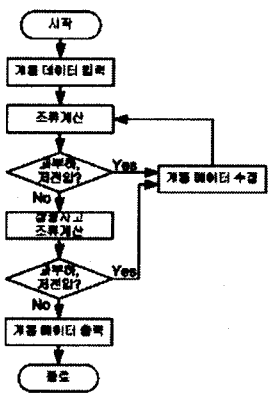
오늘날 수많은 업체들이 고객의 문제해결을 위해 노력하고 있다. 고객이 머릿속에 그리는 시스템을 컴퓨터로 구현하기 위해 여러 가지 상상과 경험을 동원하여 시스템을 구축하기도 한다. 그러한 시스템을 가능하게 하는 도구로서, 객체지향(Object-Oriented)개념 및 객체지향의 개발도구들이 등장하게 되었고 그것들을 이용하여 상당한 성과를 거두었다. 그러나 비즈니스와 IT환경의 빠른 변화 속도, 다양한 수정요구 등에 발맞추기에는 서비스 수준이 부적합한 요소가 많이 있었다. 이후 분산 객체라는 개념이 발전되었고 소프트웨어 모듈을 근간으로 소프트웨어를 개발하려는 시도가 계속되면서 점차 컴포넌트(Component)라는 개념이 더욱 중요해졌다. 특히 애플리케이션 서버(Application Server)에서 동작하는 형태나 기존 모듈에 래핑(Wrapping)한 형태의 컴포넌트는 이미 개발환경 속에 깊이 파고들어와 있는 상태이다. 컴포넌트란 "독립적인 단위기능을 수행하는 소프트웨어 모듈"을 의미하는데, 특히 CBD(컴포넌트 기반 개발, Component Based Development)[1] 패러다임을 완벽히 지원하여 블록을 조립하듯이 프로그래밍을 할 수 있는 것이 최대의 장점이다. 이는 마치 레고 블록을 조립하여 건물이나 비행기 등을 완성하는 것과 같은 방법이다. CBD 개발 방법론을 가능하게 하는 것은, 컴포넌트가 [그림 1]과 같은 다섯 가지 특징을 수반하기 때문이다.



[그림 1] 컴포넌트의 특징

그러나 이와 같이 훌륭한 도구들을 가지고도 시뮬레이션이나 실험을 수행하기 위해 컴퓨터 프로그래밍이라는 과정을 거쳐야 하는 것이 현실이다. CBD 개발 방법론이 프로그램 개발자들에게는 더 쉽고 풍부한 자원을 제공해 줬지만, 개발자가 아닌 사람에게는 여전히 높은 장벽으로 인식되는 것이 사실이다. CBD라는 개념을 이해하지 않고도 사용자의 직관과 시각에 의지하여 원하는 결과를 얻는다면 그 보다 더 좋은 일은 없을 것이다. 대부분의 프로그램은 어떠한 공정을 통해 결과를 얻게 되고, 또한 반복적으로 행해지는 편이다. 이러한 공정은 순서도(flow chart)로서 나타내는 것이 보통이다. 순서도가 주어지면 컴퓨터 프로그래밍을 통해 코딩과 디버깅을 거쳐 결과를 얻게 된다. 만약에 컴퓨터 프로그래밍 단계를 생략하고 순서도만으로 원하는 결과를 얻을 수 있다면, 사용자에게 이보다 환영할 만한 것은 없을 것이다.

본 논문에서는 이러한 사용자의 욕구를 채우기 위한 개념을 정립하고 발전시키고자 한다. 다음 [그림 2]를 살펴본다. 순서도는 전력계통 운영을 위한 간략한 절차[1]를 나타낸다. 사용자는 이러한 절차에 의해 전력계통 해석 상용 툴을 사용하여 향후 전력계통을 계획하거나 운영하고 있는데, 매우 반복적이고 시간 소모가 많은 편이다. 본 논문에서는 [그림 2]와 같은 순서도를 작성하는 것만으로, 전력계통 계획과 운영이 가능하게 하는 개념과 도구를 소개하고자 한다.



[그림 2] 전력계통 운영 절차

본 연구진이 개발한 도구는 "Smartflow"라는 도구이다. Smartflow란 독립된 기능을 수행하는 컴포넌트를 그림으로 순서에 맞게 배치하고 실행하여 사용자가 원하는 결과를 얻을 수 있게 하는 도구이다. Smartflow는 다음과 같은 특징들을 갖고 있다.

- UML의 활동 다이어그램(Activity Diagram)[2]에 기

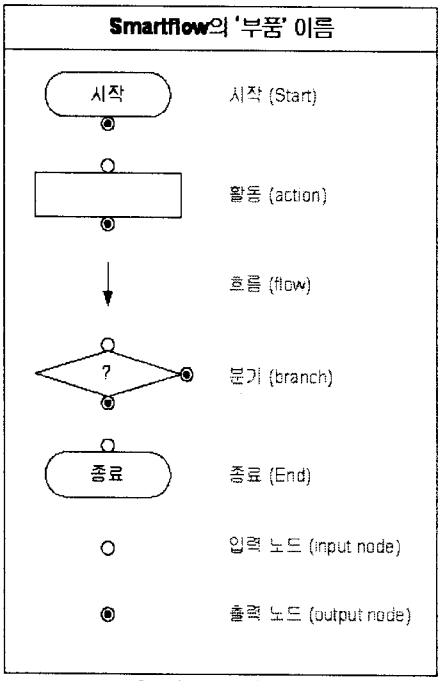
1) 단순화를 위해 계통 안정도는 포함하지 않음.

초한다.
- 활동(activity)에 중점을 둔다
- 컴포넌트의 모든 특징을 갖고 있다.
  - 미리 컴파일된 실행코드 기반
  - 용도, 유형, 인터페이스 정보 포함
  - 표준화된 기술 사용
  - 패키지화
  - 배포 가능
- 사용자의 논리를 바탕으로 하는 순서도를 작성하여 원하는 결과를 얻을 수 있게 한다.

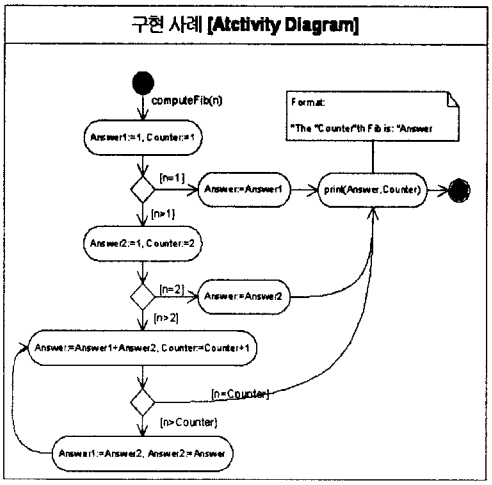
2. 본 론

2.1 구성요소

Smartflow의 구성요소를 설명한다. 구성요소의 가장 작은 단위를 '부품'이라 하였다. [그림 3]과 같이 크게 7개의 요소로서 구성된다. 시작(Start)과 종료(End) 부품은 반드시 한번만 사용되어야 하고 Smartflow의 시작과 끝을 의미하게 된다. 활동(activity)과 분기(branch) 부품은 우리가 관심을 두고 있는 문제의 영역에 따라 여러 가지 내용으로 채워질 수 있다. 전력계통에 대해서는 전력조류계산[활동]이나 과부하 체크[분기] 등 일 수 있고, 기계 시스템에서는 밸브조절[활동]이나 압력 체크[분기] 등 이 될 수 있다. 흐름(flow) 부품은 시작, 활동, 분기 부품을 연결해 주는 역할과 데이터의 시간적 흐름을 나타내게 된다. [그림 4]는 Smartflow의 구성요소를 이용하여 활동 다이어그램(Activity Diagram)을 그린 것이다. 간단한 수리연산을 수행하는 것으로서 Smartflow의 구성요소를 사용하여 나타낸 것을 알 수 있다.

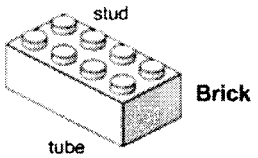


[그림 3] 구성요소



[그림 4] 활동 다이어그램(Activity Diagram)

Smartflow 부품은 몇 개의 특성을 갖는데, 첫째로, 독립성이다. 모든 부품은 하나의 부품으로도 완전한 기능을 하도록 하는 것이다. 둘째로, 결합성으로서 다른 부품과 쉽게 결합되어 데이터가 전달되게 하는 것이다. 셋째로, 일반성인데, 입출력 특성과 생성 및 활동 소멸 과정을 거친다는 것이다. Smartflow의 부품은 [그림 5]에 있는 레고의 brick과 비슷한 개념을 갖고 있다. 레고의 brick들을 잘 조합하면 성을 만들기도 하고 배를 만들 수도 있다. Smartflow의 부품도 시스템으로서 확장될 수 있는 가능성을 모두 가지고 있다. 전력 시스템에 적용하고자 한다면, 전력계통과 관련된 항목들을 분류하여 Smartflow의 부품으로 만들면 된다. 기계 시스템 적용할 때도 마찬가지로, 관련된 항목들을 분류하여 부품으로 만들면 되는 것이다. 실로 가능성이 무궁무진하다고 할 수 있겠다.

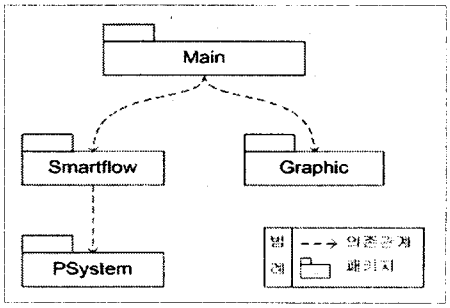


[그림 5] 레고의 brick

2.2 Smartflow 설계

2.2.1 시스템 구성 및 개발환경

개발해야할 그룹을 [그림 6]과 같이 4개의 패키지로 분류하였다. Main 패키지는 전체를 총괄하는 기능을 수행하는 것으로서 개발 전체 화면으로 나타난다. Graphic 패키지는 Smartflow를 표현하기 위해 다이어그램을 그릴 수 있도록 하는 기능을 한다. Smartflow 패키지는 다이어그램 디자인과 실행에 필요한 기능들을 수행한다. PSystem 패키지는 전력 시스템에 대해 분류된 부품들로 구성되어 있다. 여기에서 Smartflow 패키지가 처리할 수 있는 패키지는 비단 PSystem 패키지에만 국한된 것이 아님을 밝혀둔다. 여타의 다른 시스템에도 적용될 수 있다. 현재는 문제를 전력계통에 적용하는 것으로 한다.



[그림 6] 시스템 구성

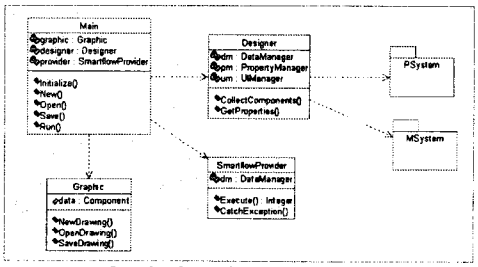
개발환경은 [표 1]과 같고, 특별히 Graphic 모듈의 처리를 위해 MS Visio를 활용하였다.

[표 1] 개발환경

구 분		내 용
개발 환경	CPU	펜티엄 4급 이상 PC
	OS	Windows 2000/XP
	개발 언어	C#
	개발 도구	MS Visual Studio .NET, MS Visio

2.2.2 클래스 다이어그램

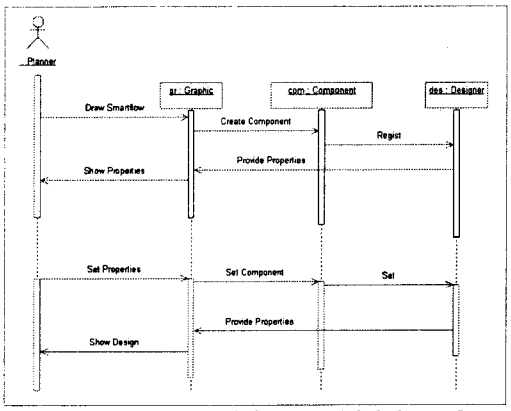
클래스 다이어그램은 시스템의 정적인 관점을 표현하고, 클래스들 간의 관계를 기술하는 것으로서, 각 클래스에는 멤버와 함수를 나타낸다. [그림 6]은 Smartflow의 클래스 다이어그램을 나타낸다. Main 클래스는 Graphic, Designer, SmartflowProvider 클래스의 객체를 멤버로 갖고 있다. 함수로는 초기화를 수행하는 Initialize()함수와 파일 입력 및 저장을 수행하는 New(), Open(), Save()함수, 그리고 시뮬레이션을 실행하는 Run()함수가 있다. Graphic 클래스는 사용자가 다이어그램을 그릴 수 있는 캔버스에 해당하는 것으로서, 그래픽 데이터를 나타내는 data 멤버를 갖고 있고, Main 클래스의 함수를 실행하기 위한 함수들로 구성되어 있다. Designer 클래스는 멤버로서, 데이터를 관리하는 기능을 수행하는 DataManager와 속성(property)를 설정하고 관리하는 기능을 수행하는 PropertyManager, 그리고 각종 대화상자를 관리하는 기능을 수행하는 UIManager를 가지고 있다. 주요한 함수로는 데이터를 수집하는 기능을 수행하는 CollectComponents()함수와 선택한 데이터의 속성 정보를 제공하는 GetProperties()함수가 있다. SmartflowProvider 클래스는 Designer 클래스에서 관리되던 DataManager 멤버를 받아 실행과 예외를 처리하는 기능을 수행한다. PSystem 패키지는 전력계통과 관련된 컴포넌트의 집합이다. MSystem은 기계시스템과 관련된 컴포넌트의 집합을 의미한다.



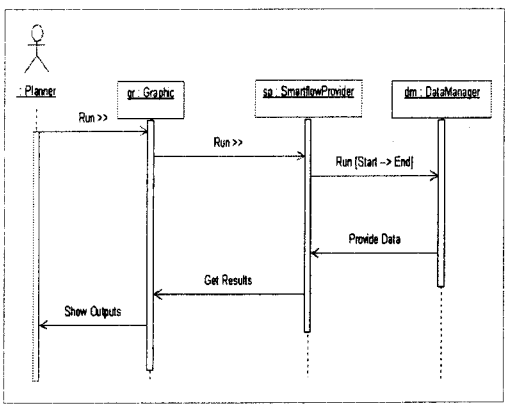
[그림 7] 클래스 다이어그램

2.2.3 시퀀스 다이어그램

시퀀스 다이어그램은 시간에 따른 객체 사이의 메시지 발생순서를 강조하며, 시간의 흐름에 따라 메시지들을 위에서 아래로 세로축에 따라 배치하는 다이어그램이다. Smartflow에서는 사용자(Planer)가 적절한 컴포넌트를 화면에 배치하는 디자인 모드와 디자인된 결과를 실행하여 결과를 확인할 수 있는 실행 모드로 구분하여 시퀀스 다이어그램을 나타낼 수 있다. [그림 8]은 디자인 모드에서의 시퀀스 다이어그램으로서, 상단에 위치한 막대는 사용자가 컴포넌트를 배치할 때 동작하는 객체와 메시지를 나타낸다. 하단에 위치한 막대는 컴포넌트에 속성(property)을 부여하고자 할 때 기인하는 객체와 메시지를 나타낸다. [그림 9]는 실행 모드에서의 시퀀스 다이어그램으로, 사용자가 실행 명령을 내리고 결과를 얻는 시간적인 메시지의 흐름을 나타낸다.



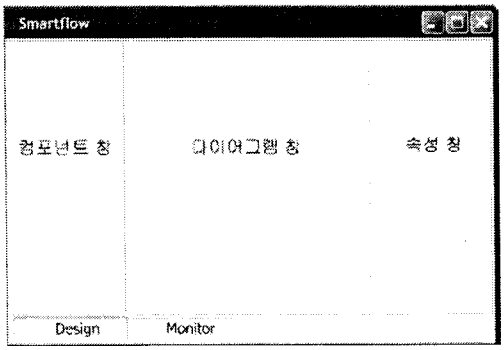
[그림 8] 시퀀스 다이어그램 - [디자인 모드]



[그림 9] 시퀀스 다이어그램 - [실행 모드]

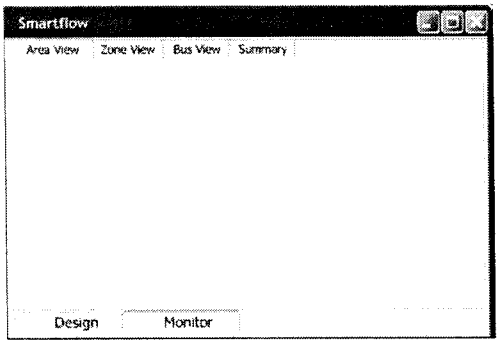
2.2.4 화면 설계

시퀀스 다이어그램에서 밝혔듯이 Smartflow는 디자인 모드와 실행 모드로 구분된다고 하였다. 화면에서도 이와 같은 원칙이 적용되어 디자인 창과 모니터 창으로 나뉘게 된다. [그림 10]은 디자인 창으로서 컴포넌트 창, 다이어그램 창, 속성 창으로 구성되어 있다. 컴포넌트 창은 사용자가 사용할 수 있도록 등록된 컴포넌트 리스트를 나타낸다. 다이어그램 창은 컴포넌트 창에 있는 컴포넌트를 마우스로 끌어 배치할 수 있도록 하는 창으로서 [그림 2]와 같은 다이어그램을 그릴 수 있게 한다. 속성 창은 다이어그램 창에 배치된 컴포넌트의 속성을 설정할 수 있게 하는 창이다.



[그림 10] 화면 설계 - 디자인 창

[그림 11]은 모니터 창으로서 실행 모드를 수행한 후의 결과를 볼 수 있도록 하는 창으로서, 다이어그램 창에 배치한 컴포넌트에 의해 좌우된다. 예로서 조류계산을 수행하였을 경우에는 계통 요약정보와 선로 조류 등이 모니터 창에 표시될 수 있는 항목들이 될 수 있다.

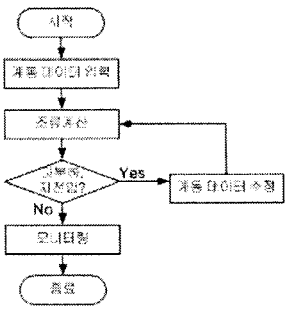


[그림 11] 화면 설계 - 모니터 창

2.3 사례 연구

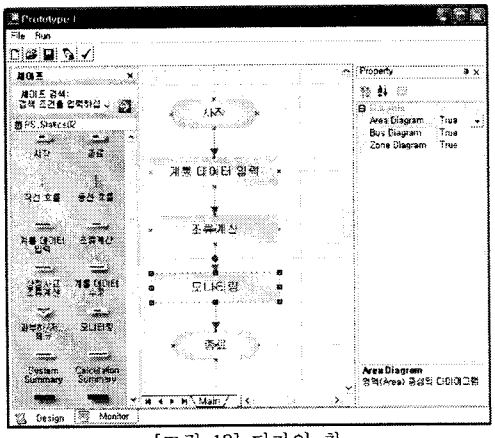
2.3.1 전력 계통운영 방안 적용

본 연구에서 실현하고자 하는 것은 [그림 2]와 같은 다이어그램을 그려서 프로그램을 자동으로 실행하도록 하는 것이다. 그러나 현 단계에서는 새로운 개념을 도입하여 체계화시키고, 자동화의 가능성을 테스트하고자 하는 것이 주요 목적이기 때문에 원래의 문제를 단순화 시켜 [그림 12]와 같이 간략화 하였다.



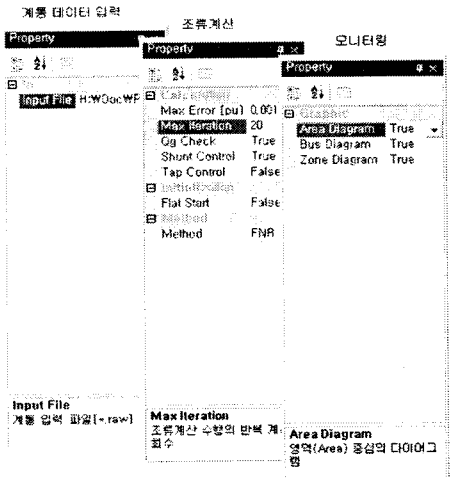
[그림 12] 간략화된 전력계통 운영 프로그램

[그림 12]를 프로그램에 적용한 것이 [그림 13]이다. 총 5개의 컴포넌트와 흐름(flow) 컴포넌트가 준비되어야 하는데 [그림 13]의 좌측 '컴포넌트 창'에 이러한 컴포넌트가 준비되어 있는 것을 확인할 수 있다. 가운데의 '다이어그램 창'에는 디자인된 컴포넌트들이 배치되어 있는 것을 확인할 수 있으며, 우측 '속성 창'에 모니터링 컴포넌트에서 보여줄 대상으로 Area Diagram, Bus Diagram, Zone Diagram이 모두 선택된 것을 확인할 수 있다.



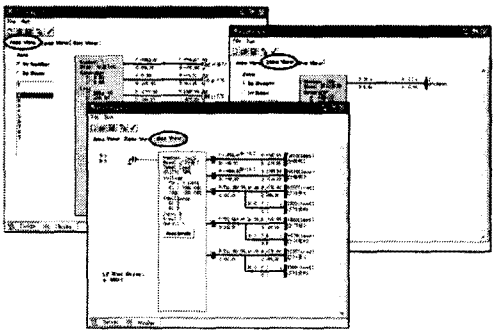
[그림 13] 디자인 창

[그림 14]는 각 컴포넌트 선택시의 속성 창에 나타나는 내용을 각각 나타낸 것이다. 속성 창에서는 컴포넌트의 속성을 마음대로 변경할 수 있게 한다. 컴포넌트의 모든 속성은 기본값(default)과 설명을 가지고 있어 사용자에게 편리한 정보를 제공한다. 컴포넌트의 속성 설정은 상당히 중요한 개념으로서 같은 다이어그램을 그렸다 하더라도 주어진 속성을 달리 할 경우 여러 가지 결과를 얻을 수 있게 한다.



[그림 14] 컴포넌트 속성

사용자가 해야 할 일은 여기까지 이다. 간단히 마우스로 컴포넌트를 몇 개 배치하고 속성을 설정하면 원하는 결과를 얻을 모든 준비가 끝난 것이다. 이제 [그림 13]에서 [Run] 메뉴를 실행하면 [그림 15]와 같이 모니터 창에 결과가 나타난다. 모니터 창에서는 조류계산 수행 후의 조류량을 표시하는 것으로서 각각 영역(Area), 지역(Zone), 모선(Bus) 중심의 조류 흐름을 나타내는 것을 확인할 수 있다.



[그림 15] 실행 결과 - 모니터 창

2.3.2 각 프로그램 비교

Smartflow의 특징을 비교해 보기 위해 전력계통 해석 상용 툴로서 많이 사용되고 있는 PSS/E의 IPLAN[3]과 비교를 수행하였다. IPLAN은 PSS/E와 같은 응용 프로그램을 더욱 효과적으로 사용하기 위해 개발된 프로그래밍 언어이다. 그러나 IPLAN은 사용자 인터페이스에 있어 PSS/E 프로그램에 있는 함수를 호출하여 사용하도록 하고 있어 사용자에게 부담을 가중시키고 있는 형편이다. 항목 별 비교 결과는 [표 2]와 같다. 대부분의 경우에 Smartflow가 우위를 점하는 것을 알 수 있다. 이용 가능한 리소스가 부족하다는 측면이 있지만, 이는 Smartflow를 계속 발전시키면서 극복될 수 있는 경미한 사항이다.

[표 2] 상용 프로그램과의 비교

구 분	Smartflow	IPLAN [PSS/E]
기본 원리	활동 다이어그램 이론	프로그래밍 방식
사용자 인터페이스	그래픽 다이어그램	프로그래밍
사용 편리성	우수	난해[프로그래밍 습득]
자동화	가능	가능[프로그래밍 필요]
학습 속도	빠르다	느리다
이용 가능한 리소스	적다	많다

3. 결 론

소프트웨어 개발사에 있어 과거의 절차 중심적인 언어로부터 객체지향 방법(OOP), 컴포넌트 기반 개발 방법(CBD)에 이르기 까지 많은 발전이 있어 왔다. 그로인해 개발자에게는 더 많은 기회와 풍부한 자원을 제공된 이 사실이다. 그러나 최신의 개발 방법론이 있음에도 사용자로서는 여전히 반복적이고 시간 소모적인 일을 맡을 수밖에 없게 된 것이 지금까지의 현실이다. 이제는 사용자도 풍요로움을 누릴 때가 되었다. 사용자가 복잡한 절차를 수행하지 않고 직관적인 다이어그램을 그리는 것만으로 원하는 결과를 얻을 수 있게 해야 되는 것이다.

이를 위해 본 연구진은 활동 다이어그램에 근거하여 독립된 기능을 수행하는 컴포넌트를 그림으로 순서에 맞게 배치하고 실행할 수 있게 하는 Smartflow 라는 도구와 개념을 발전시켰다. 특히 컴포넌트의 속성을 설정할 수 있게 하는 기능은 컴포넌트의 다양성을 부여한다는 것을 알 수 있게 하였다. 그리하여 같은 다이어그램이라 하더라도 컴포넌트에 속성을 어떻게 부여하는가에 따라 다양한 결과를 얻을 수 있다는 것을 알 수 있었다. 사례 연구를 통해 컴포넌트 디자인이 매우 간결하고 직관적이라는 것을 알 수 있었고 결과를 확인할 수 있었다. 또한 현존하는 전력계통 해석 도구와의 비교를 통해 이용 가능한 자원이 부족하다는 것을 제외하고는 상당히 우위를 점하고 있다는 것을 알 수 있었다. 현재 기초기술 단계에 있지만, 점차 응용기술과 상용기술 단계로 나아가게 되면 전력계통 운영의 자동화도 그리 요원한 일은 아닌 것이다.

Smartflow의 활용 분야로서 전력계통 운영, 전력수급 운영, 전력거래 등을 고려할 수 있다. 그 외에도 기계시스템과 같은 여타의 다른 시스템에도 널리 활용이 가능하다.

[참 고 문 헌]

[1] 컴포넌트비전(주), "실전 CBD Project", 영진닷컴, 2004

[2] Martin Fowler, "UML Distilled", Addison Wesley Longman, 2000

[3] PSS/E-28 IPLAN Program, Power Technologies, Inc. 2001

[4] PRABHA KUNDUR, "Power System Stability and Control", McGraw-Hill, 1994

[5] Erich Gamma, John Vlissides, Ralph Johnson, Richard Helm, "Design Patterns : Elements of Reusable Object-Oriented Software ", McGraw-Hill, 1995

[6] Brian Johnson, Craig Skibo, Marc Young, Inside Microsoft Visual Studio .NET, Microsoft Press, 2003