

#### 저작자표시-비영리-동일조건변경허락 2.0 대한민국

#### 이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

#### 다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우 에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건 을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer





## 공학박사학위논문

# 연결, 소통, 복제 기능이 강화된 농업시스템응용플랫폼의 개발

Development of Agricultural Systems

Application Platform enhancing Connection,

Communication, and Cloning

2014 년 8 월

서울대학교 대학원 생태조경·지역시스템공학부 김 태 곤

# 연결, 소통, 복제 기능이 강화된 농업시스템응용플랫폼의 개발

Development of Agricultural Systems Application Platform enhancing Connection, Communication, and Cloning

지도 교수 이 정 재

이 논문을 공학박사 학위논문으로 제출함 2014 년 6 월

서울대학교 대학원 생태조경·지역시스템공학부 지역시스템공학전공 김 태 곤

김태곤의 공학박사 학위논문을 인준함 2014 년 7 월

위 육	실 장	(인)
부위	원장	(인)
위	원	(인)
위	원	(인)
위	원	(인)

## 초 록

시스템 시뮬레이션은 구성요소의 거동을 모델링하고, 구성 요소간의 관계를 정의함으로써 구현된다. 일반적으로 관계를 정의하는 과정에서 복잡도가 증가되어 시스템 시뮬레이션 모델을 구성하기가 어렵다. 복잡도 문제를 해결하기 위하여 절차적 방식의 프로그래밍 패러다임은 객체지향 프로그래밍 패러다임으로 발전하였다. 그러나 객체지향 패러다임을 이용하여 개발된 많은 연구들이 여러 요소를 결합하는 과정에서 절차적 방식으로 환원되어 결과적으로 독립성은 저해하고, 여전히 복잡도를 해결하지 못하였다.

이러한 문제를 해결하기 위하여 다양한 시스템 시뮬레이션 도구들의 기능을 검토하고, 복잡하면서도 성공적인 시스템인 생명체의 거동을 분석하여 시스템 시뮬레이션 도구가 갖추어야 할 기능을 도출하였다. 모델의 독립성을 유지하고, 복잡도를 낮추어시스템 시뮬레이션을 수행할 수 있는 범용 농업시스템 시뮬레이터 (Generic Agricultural Systems Simulator, GASS)의 단위 객체의독립성을 유지하기 위한 일방향 정보전달기법을 주요한 기능으로 채택하였다. 또한 생명체가 생명을 유지하는 과정에서 다음 3가지거동 특성을 갖는다고 분석하였다. 첫째, 시스템을 구성하는 요소는기본적으로 인접한 요소 간에 정보를 교환하지만, 일부는 거리와무관하게 연결된다. 둘째, 같은 구성요소라도 소속 그룹에 따라소통의 행태가 달라진다. 셋째, 시스템의 구조는 거시적으로 고정된형태를 지니지만, 요소의 복제, 소멸을 통하여 경계가 지속적으로 변화한다.

본 연구에서는 앞서 도출한 기능을 바탕으로, 3가지 주요 특징인 연결성(Connection), 소통구조(Communication), 복제기능(Cloning), 즉 3C을 강화한 농업시스템응용플랫폼(Agricultural Systems Application Platform, ASAP)을 개발하였다. 네트워크를 이용하여 인접성에 기반한 물리구조와 위상관계에 기반한 정보구조를 동시에 표현하고, 동질소통구조와 이질소통구조를 모두 반영할 수 있도록 프로그램을 설계하였으며, 컴포넌트의 생성 및 사멸을 통하여 시스템의 경계가 변하는 문제에 적용 가능하도록 시뮬레이션 플랫폼을 개발하였다. 또한 개발환경에 대한 기본 프로토콜을 제시하고, 모델 개발자를

위한 컴포넌트 작성 프로토콜 및 연구자를 위한 컴포넌트 활용 프로토콜을 설명하고, 사용자 편의를 고려한 유틸리티를 기술하였다.

개발된 농업시스템응용플랫폼의 적용성을 검토하고자 다양한 지역 시스템 공학 문제를 풀이하였다. 모형의 적용은 기존 모델과 비교하여, 간단한 모델 구현과 시뮬레이션 구동단계를 거쳐 동일한 분석 결과를 확인함으로써 검증하였다. 연구 결과를 바탕으로 수학적, 공학적 기법을 이용한 하향식 모델링 방식에서 시뮬레이션 기법을 통한 상향식 모델링 방식이 가능하였다. 향후 개발한 컴포넌트의 배포를 진작하여, 모델 갱신과 지식 공유를 통한 지역시스템 공학의 발전을 기대한다.

주요어: 연결, 소통, 복제, 객체 지향, 시스템 시뮬레이션, 플랫폼

학 번:2007-30313

## 목 차

제	1 장 서 _ 론	. 1
·	1.1 연구 배경	
	1.2 연구 내용 및 범위	
	1.3 논문 구성	
-5		
제	2 장 시스템 시뮬레이터	. 5
	2.1 지역시스템과 시스템 시뮬레이터	5
	2.2 다양한 시스템 시뮬레이션 도구	6
	2.3 GASS	8
	2.3.1 개요	8
	2.3.2 인접성	10
	2.3.3 속성정보 관리	
	2.4 시스템 시뮬레이션 도구의 한계	12
제	3 장 ASAP의 개발	17
	3.1 발전된 시뮬레이션 프로토콜	
	3.1.1 연결 프로토콜	17
	3.1.2 소통 프로토콜	18
	3.1.3 복제 프로토콜	19
	3.2 시뮬레이터의 설계	
	3.2.1 컴포넌트 개발 프로토콜	21
	3.2.2 컴포넌트 이용 프로토콜	
	3.2.3 시뮬레이션 유틸리티	23
	3.3 연결성에 대한 검토	
	3.3.1 물리구조로 연결된 시스템	
	3.3.2 위상구조로 연결된 시스템	
	3.3.3 혼합 시스템	
	3.4 소통구조에 대한 검토	
	3.4.1 동질소통구조에서의 속성	
	3.4.2 이질소통구조에서의 속성	
	3.5 복제 기능에 대한 검토	
	3.5.1 복제 기능을 통한 프랙탈 구조 재현	
	3.5.2 네트워크 시스템의 전처리	
	3.6 ASAP의 구현	37
	3.6.1 시뮬레이션 환경의 구현	
	3.6.2 시뮬레이션 모델의 구현	39

제 4 장 지역시스템 문제의 적용	41
4.1 서론	
4.2 모델 구현과 구동	
4.3 동질소통을 하는 위상구조	45
4.3.1 문제 선정	
4.3.2 연결된 저수조의 수위해석	
4.3.3 삼형제 저수지의 수위 모의	
4.4 동질소통을 하는 물리구조	
4.4.1 문제 선정	
4.4.2 연직배수를 고려한 1차원 토질 압밀	61
4.4.3 점적 관개에 의한 토양수분 모의	
4.4.4 배수갑문 염해 모의	77
4.5 이질소통을 하는 위상구조	85
4.5.1 문제 선정	85
4.5.2 전염확률을 고려한 감염농가 확산	85
4.6 이질소통을 하는 물리구조	85
4.6.1 문제 선정	92
4.6.2 트러스 구조의 해석	
4.7 복제기능을 통해 성장하는 시스템	
4.7.1 문제 선정	104
4.7.2 초목의 성장모의	104
4.7.3 전처리 기능을 활용한 등가트러스 해석	107
4.8 소결	109
제 5 장 결 론	110
5.1 요약 및 결론 5.2 향후 연구과제	
5.2 양후 연구과제	114
참고문헌	115
Abstract	121

## 표 목차

Table 1 Comparison of GASS components and cells	14
Table 2 Reserved words for ASAP	22
Table 3 Classification of example problems	41
Table 4 Watershed characteristics of three reservoirs	57
Table 5 Level-storage (m- $10^3$ m $^3$ ) lookup table for three reservoirs	58
Table 6 Soil hydraulic parameters for textural class (Choi et al., 2012)	74
Table 7 Properties of nodes and edges for 5-truss problem	98
Table 8 Displacements comparison of ANSYS and this study	101

## 그림 목차

Figure 1 SIMILAR for system engineering	5
Figure 2 Concept of unidirectional information fetch scheme (Yi, 2003).	9
Figure 3 Schematic view of established simulation model (Yi, 2003)	11
Figure 4 Concept diagram of homogeneous structure for communication	on19
Figure 5 Concept diagram of heterogeneous structure for communicati	on .19
Figure 6 Class diagram of main class and component interface	21
Figure 7 Problem of 2D heat conduction (Kim, 2007)	26
Figure 8 Results of 2D heat conduction problem using ASAP	27
Figure 9 Results of Stochastic PERT	27
Figure 10 Network view for heat conduction with controller	29
Figure 11 Result of grid temperature with controller	30
Figure 12 Problem of truss member force	31
Figure 13 Simulation viewer to solve truss member force	32
Figure 14 Problem for moment distribution method	33
Figure 15 Solution for moment distribution method	33
Figure 16 A main code for parallel communication	34
Figure 17 Simulation viewer to solve moment distribution problem	34
Figure 18 Simulation for lindenmayer systems	35
Figure 19 Class diagram related with simulator environment	38
Figure 20 Class diagram related to implement system models	40
Figure 21 A main code for setting simulation environment	44
Figure 22 Problem for water height function of one tank	46
Figure 23 Problem for water height function of two tanks	48
Figure 24 Problem for water height function of three tanks	49
Figure 25 Program code for water height problem with linked tanks	51
Figure 26 Screenshots of ASAP for two tanks problem	52
Figure 27 Comparison of simulated result to analytic result	53
Figure 28 Simulation model results related to delta time	54
Figure 29 Simulated water height for three tanks problem	55
Figure 30 Map of three reservoirs	56
Figure 31 Simulated results of weather and watershed components	58
Figure 32 Water level simulation for linked three reservoirs	59
Figure 33 Simulated water level among the linked three reservoirs	60

FIGURE 34 SCATTER PLOT BETWEEN OBSERVED AND SIMULATED LEVEL	61
FIGURE 35 DEPTH-TIME GRID FOR SIMULATING CONSOLIDATION	65
Figure 36 A main code of consolidation component	66
FIGURE 37 RESULTS OF CONSOLIDATION PROBLEM USING ASAP	66
Figure 38 A main code of distribution component for soil water	73
Figure 39 Simulated results of soil moisture distribution (4hr irrigation)	76
FIGURE 40 EXPLICIT FINITE DIFFERENCE METHOD	80
FIGURE 41 A MAIN CODE OF CHLORIDE PENETRATION COMPONENT	80
Figure 42 Modelling for Chloride attack of Marine Concrete Structure	81
Figure 43 Chloride attack progress on the corner at 1st year, $10$ years lat	ΓER,
and 20 years later	82
Figure 44 Chloride attack progress at a 10cm deep point from surface	
WITHOUT REPAIR OR RECOVER TREATMENT	82
Figure 45 Chloride attack and repair progress at 19 years later, 20 years	
LATER, AND 21 YEARS LATER	83
Figure 46 Chloride attack progress considering repair every 20 years at a	
10cm deep point from surface without repair or recover treatment	84
Figure 47 A main code of infection component	87
Figure 48 Initial condition of Network model for HPAI infection	88
Figure 49 Link weights for infection in 39 nodes	89
Figure 50 Simulated Network about infection status of Stockbreeding	90
FIGURE 51 NUMBER OF INFECTIVE STOCKBREEDING	91
Figure 52 A basic form and a unit component of 2D truss structure	93
FIGURE 53 A MAIN PROGRAM CODE FOR TRUSS NODE COMPONENT	96
Figure 54 Ten bar problem and solution (This study)	97
FIGURE 55 PROBLEM SET FOR 5-TRUSS STRUCTURE	97
Figure 56 A main program to set up problem	99
FIGURE 57 RESULTS OF 5-TRUSS PROBLEM USING ASAP	100
FIGURE 58 EXAMPLE FOR 11-TRUSS STRUCTURE	101
Figure 59 Flow chart of FEM and truss simulation with ASAP	102
Figure 60 A main program code for plant growth	105
FIGURE 61 SIMULATION FOR PLANT GROWTH BASED ON L-SYSTEMS	107
Figure 62 Simulated internal energy for deep beam to solve by equivalent	
TRUSS COMPONENT	108

## 제 1 장 서 론

## 1.1 연구 배경

지역시스템공학은 지역의 자원 및 공간을 효과적으로 활용하기 위하여 시스템 공학을 적용하는 학문으로, 시스템을 구성하는 요소와 구성요소 간의 상호작용을 분석하는 시스템 공학적 과정이 중요하다. 시스템공학은 문제를 해결하기 위해서 대안을 분석하고, 시스템을 모델링하여, 요소를 결합하고, 구성된 시스템을 구동하여, 그 결과를 평가하는 과정으로 이루어진다(Bahill and Gissing 1998). 이러한 과정을 지역시스템 분야에 접목하기 위해서는 연구자가 쉽게 모델을 구성하고 모의할 필요가 있어 시스템 시뮬레이션 분야가 주목을 받게 되었다.

시스템 시뮬레이션 모델을 만들기 위한 노력은 일반적으로 심각하게 복잡하고, 활용분야로 한정되거나, 유지보수가 어려운 모델이 되기 쉽다(Jones et al. 2001). 이런 어려움을 해결하고자 많은 연구자들은 객체지향 프로그래밍 패러다임과 같은 소프트웨어 공학 기술을 이용하여 문제를 해결하고자 노력하였다(Acock and Reddy 1997; Van Evert and Campbell 1994).

객체지향 패러다임의 도입으로 캡슐화를 통한 접근제어, 상속을 통한 코드 재사용성 향상, 다형성을 통한 코드의 동적결합 등을 성취하고, 코드의 유지관리가 가능하다고 판단하였다. 그러나다수의 서브 시스템이 결합하는 과정에서 여전히 절차적 패러다임을 고수하고 있기 때문에, 시스템 간의 정보 전달을 효과적으로 정의하기 어려우며, 특히 시스템이 확장될수록 시스템사이의 연결 수가 급격히 증가하여 궁극적으로 시스템 통합이불가능하였다(이한귀 et al. 2012).

이러한 요소들 간의 연결 복잡성을 해결하기 위하여 시스템을 구성하는 서브모델 사이의 정보전달체계를 단순화한 일방향 정보전달 기법(Unidirectional Information Fetch, UIF)이 제안되었다(Yi 2003). 일방향 정보전달 기법을 이용하여 서브시스템 사이에 양방향으로 정보를 주고받는 방식에서 탈피하여, 인접한 시스템에 원하는 정보를 가져올 수만 있고, 주입할 수

없도록 설계한 범용 농업 시스템 시뮬레이터(Generic Agricultural Systems Simulator, GASS)가 개발되었다. 편의성을 높이기 위해 컴포넌트 프로그래밍 패러다임을 시각 프로그래밍에 적용하여 사용자가 시스템을 도식적으로 구성하도록 하였다.

GASS는 개별 시뮬레이션 모델을 상징하는 사각형 심볼을 화면상에서 인접하게 위치시킴으로써 맞닿은 모델끼리 서로 연결되었음을 정의하였다. 위치 정보를 이용하여 연결 정보를 표현함으로써 시스템을 매우 효과적으로 표현하였다. 그러나, 서로 멀리 떨어진 요소들끼리 연결할 수 없다는 연결성(connection)의 제약, 모델을 맞닿아 연결함으로 인하여 네트워크 구조 혹은 연결선 개념을 표현하기 어렵고, 연결선에 대한 속성을 입력할 수 없는 소통구조(communication)의 제약, 시스템의 외연이 변화하는 현상을 모의할 수 없는 복제기능(cloning)의 제약이 있었다.

본 연구에서는 GASS를 근간으로 하여, 다양한 시스템 시뮬레이션 도구가 갖는 한계를 극복하고, 편의성을 고려하여 농업시스템응용플랫폼(Agricultural Systems Application Platform, ASAP)를 개발하였다. 개발한 시스템 시뮬레이션 플랫폼은 다양한 지역시스템 문제를 적용하여 효용성을 증명하고자 하였다.

## 1.2 연구 내용 및 범위

본 연구는 시스템 시뮬레이션 도구를 검토하고, 지역시스템 분야에 적용하기에 적합하도록 Yi (2003)가 제안한 GASS를 기반으로 3가지 측면에서 개선한 ASAP를 개발하였다. 개선된 3가지 기능은 연결성(connection), 소통구조(communication), 복제기능(cloning)으로, 이러한 3C 기능에 대하여 예제를 통해 적합성을 검토하였다. 또한 편의성 향상을 도모하기 위하여 개발된 시뮬레이션 플랫폼의 기능에 대하여 기술하였다.

개발된 ASAP은 분야별로 정리한 지역시스템의 다양한 문제를 적용함으로써 활용 가능성을 평가하였다. 모의하고자 하는 문제는 3C 관점에서 편중되지 않게 분포하도록 선정하였으며, 각 예제는 메커니즘을 설명하는 지배방정식과 이를 컴포넌트로 구현하는 방법, 구현된 컴포넌트를 이용하여 시스템 시뮬레이션을 수행한 분석 결과를 각각 서술하였다.

### 1.3 논문 구성

본 논문은 5개의 장으로 구성되며, 각 장은 다음과 같은 내용으로 구성된다.

제 1장에서는 본 연구의 배경 및 목적에 대해서 기술하였다.

제 2장에서는 시스템 시뮬레이션에 관한 선행 연구를 조사하여 분석하고, 시스템 시뮬레이션에서 필요한 요구사항을 도출하였다. 시스템 시뮬레이션에 있어서 복잡도 증가를 억제하고, 시스템의 자기창발을 모의할 수 있는 도구로 GASS를 소개하였다. 다양한 시스템 시뮬레이션 도구가 소개되었고, 특히 본 연구의 근간이 된 GASS는 많은 장점을 가지고 있지만, 농업시스템을 모의하는 데 있어서 연결(connection), 소통(communication), 복제(cloning) 3C 측면에서 한계가 있었다. 시스템을 구성하는 요소 간의 연결 속도 차이에 따른 기하구조와 위상구조를 동시에 갖는 시스템을 모사하기 위한 연결 측면, 요소 간의 소통구조가 동일하지 않는 이질소통구조를 갖는 시스템의 소통 측면, 경계 및 구조가 변화하는 시스템이 갖는 복제 측면에 대하여 개선 필요성을 기술하였다.

제 3장에서는 앞서 지적한 시스템 시뮬레이션 도구의 한계를 극복하기 위하여 3C 측면에서 새로운 프로토콜을 제시하고, 이를 바탕으로 ASAP를 개발하였다. 개발한 시뮬레이터에 대하여 각프로토콜 별로 예제를 통하여 검토하였다. 연결성에 대해서는물리구조적으로 연결된 열전도 해석 문제와 위상구조로 연결된PERT, 그리고 혼합된 문제로 특정 지점의 온도 조건에 따라변화하는 제어기가 설치된 열전도 해석 문제를 제시하였다.소통구조 측면에서는 동질소통구조와 이질소통구조에 대하여기술하고,동질소통구조는 절점법을 이용한 트러스의 해석 문제를,이질소통구조는 모멘트 분배법을 이용한 구조해석 문제를풀이하였다.마지막으로 복제 기능 측면에서는 자기 복제를 통해생장하는 식물을 모의한 프랙탈 기반의 식물생장 모형인 Lindenmayer systems을 구현하였다.

제 4장에서는 개발된 ASAP를 이용하여 다양한 분야에 대하여 적용사례를 제시하였다. 적용사례는 3C 측면을 고려하여 분류하였으며, 지배방정식과 ASAP 컴포넌트를 개발하기 위한 주요 코드를 제시하고, 작성한 컴포넌트를 이용하여 시뮬레이션을 수행하고, 기존 연구사례와 비교 고찰하였다. 동질 소통을 하는 위상구조 시스템으로 연결된 저수조의 수위해석과 삼형제 저수지의 수위를 모의하였다. 연결된 저수조의 수위해석은 수리 분야 문제로, 오리피스 공식을 이용하여 수두차에 의한 통수량을 결정하며, 시뮬레이션한 수치해는 미분방정식의 해석해와 비교하였다. 삼형제 저수지는 연결된 저수조 예제의 실제적 예제로, 충북 음성 지역의 연결된 3개의 저수지의 수위변화를 모의하여, 실측한 저수지 수위와 결과를 비교하였다. 동질소통을 하는 물리구조는 일반적으로 유한차분법으로 풀이하는 문제이며, 기존 GASS가 대상으로 한 문제영역이며, 토질, 관개, 재료분야 문제를 선정하여 풀이하였다. 토질분야는 연직배수를 고려한 1차원 토질압밀 모의, 관개분야는 점적 관개에 의한 토양수분 모의, 재료분야는 배수갑문의 역화이온 확산을 모의하였다. 이들 예제를 통하여 유한 차분법으로 정식화된 모형을 손쉽게 시스템 시뮬레이션 환경으로 구현할 수 있음을 서술하였으며, 시간 흐름에 따라 변화하는 형상을 시각적으로 도시하여 연구자가 모의결과를 직관적으로 파악할 수 있도록 하였다. 이질소통을 하는 위상구조로 전염확률을 고려한 감염농가 확산 모델을 모의하였다. 방재분야에 해당하는 문제로, 잠복기를 갖는 현상에서 극치가 지연되어 나타나는 행태를 모의하고, 컴포넌트의 소멸 과정을 통하여 국면이 진정되는 모습을 확인하였다. 이질소통구조를 하는 물리구조로 트러스 구조해석을 수행하였으며, 구조분야에 해당하는 문제를 모의함으로써 기존 해석 기법과의 차이를 비교하고, 컴포넌트 기반의 시스템 시뮬레이션의 특징을 서술하였다. 마지막으로 복제기능을 통해 성장하는 시스템으로 초목의 성장과정을 모의하고, 전처리 기능으로 활용하여 다수의 컴포넌트가 연결된 등가 트러스를 구성하고 해석하였다. 이상에서 살펴본 적용사례를 기반으로. 향후 컴포넌트의 확장을 통하여 시스템 경계의 확장 및 시뮬레이션 결과를 이용한 계획/관리 시나리오 적용에 활용 가능할 것으로 기대한다.

제 5장에서는 3C 측면에서 개선된 ASAP의 특징에 관하여 요약하고, 본 연구에서 제시한 다양한 적용 예제를 종합적으로 정리하였다. 또한 연구의 한계를 검토하고, 향후 활용 방안을 제시하여 후속 연구의 토대를 제시하고자 하였다.

## 제 2 장 시스템 시뮬레이터

## 2.1 지역시스템과 시스템 시뮬레이터

지역시스템은 지역 자원과 공간을 인간에게 유용하게 사용하는 데목적이 있는데, 지역 자원 발굴에 있어서 생산 과정이 오래 걸리고, 생산 과정에서 다양한 요인들이 복합적으로 영향을 미치며, 지역 공간은 인위적으로 변화를 가하기 어렵다. 이러한 환경은 공산품을 대상으로 대량으로 생산체계가 갖추어진 산업공학과는 달리, 인위적으로 환경을 조절하고, 반복실험을 수행하는 직접 실험으로 연구하기 어려우며, 시뮬레이션을 통한 간접 실험으로 연구를 진행해야 한다.

시스템 공학은 시스템을 해석하여, 이해관계자를 만족시킬 수 있는 최적해를 찾는 과정이다(Hazelrigg 1996). 다양한 요소가서로 영향을 주고 받는 시스템을 해석하기 위한 절차는 Figure 1과같이 SIMILAR라는 일곱 단계로 정의된다(Bahill and Gissing 1998). 지역시스템공학은 지역시스템을 시스템 공학의 과정으로풀이하는 것으로 정의할 수 있고, SIMILAR를 간접 실험, 즉시뮬레이션을 통하여 해석하는 과정으로 치환할 수 있다.

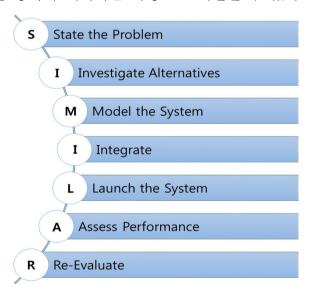


Figure 1 SIMILAR for system engineering

시뮬레이션을 이용하여 시스템을 해석하고, 최적해를 찾는 과정을 수행하기 위해서 앞서 제시한 해석 절차를 수행할 수 있는 도구가 필요하다. 시뮬레이션 도구는 현상을 모델링하고, 모델을 결합하고, 시스템의 거동을 모의하고, 모의된 결과를 평가하여, 모델의 속성을 수정할 수 있어야 한다. 이러한 과정을 지원하기 위하여 다양한 시스템 시뮬레이션 도구가 소개되었다.

## 2.2 다양한 시스템 시뮬레이션 도구

농업 시스템 시뮬레이션 도구는 소프트웨어 공학의 발전으로 통합가능한 환경을 몇몇 연구에서 제안하고 있다. 그러나 개발자들이기대하는 바와 달리 지역 시스템 공학 분야에서는 널리 활용되고있지 못하고 있는 실정이다. 이는 전통적인 모델링 접근방식에기인한 문제로 객체지향 패러다임을 기술적으로만 적용할 뿐내부적으로는 절차적 패러다임을 적용하고 있기 때문이다(Yi 2003;이한귀 et al. 2012). 전통적인 접근 방식은 'main' 모듈에서필요한 모듈을 모두 추가하고, 매개변수들을 서로 연결하여 상호정보교환을 하게끔 통합 모델을 개발한다. 이러한 방식은 main모듈을 작성하는 연구자가 개별 모듈에 대한 이해도가 높아야만가능한 작업이다.

시스템 시뮬레이터를 개발하기 위하여, 관련 연구를 조사하였다. 일반적으로 시스템 시뮬레이션이라는 용어는 전자기 시스템 시뮬레이션, 행위자 기반의 시스템 시뮬레이션, 시스템 다이내믹스 중 하나를 의미한다. 각 분야 별로 시뮬레이션 도구 및 연구흐름을 조사하고, 시스템 시뮬레이터에 필요한 요건을 도출하였다.

가장 널리 알려진 시스템 시뮬레이션 도구로는 LabView와 Simulink가 있다(손혜영 2012; 임종수 2013). 두 도구는 디지털 신호 처리에 특화된 시스템 시뮬레이터로서, 전통적인 프로그래밍 방식 대신 그래픽 환경을 제공하여 전기회로 설계에 널리이용되며(Kehtarnavaz and Gope 2006), 실험실에서 전기회로를 설계하듯 가상으로 전기회로를 구축하여 모의실험이 가능한 환경을 제공한다. 강력한 시뮬레이션 기능도 중요하게 작용하지만, 그래픽 환경을 통한 편리한 사용환경으로 많은 연구자가 이용하는 것으로 판단된다(Whitley and Blackwell 2001).

복잡계 연구에서 파생한 행위자 기반 모델링은 폰 노이만이 개념을 소개한 이래, 콘웨이와 스테판 울프람의 성과를 통하여, 셀룰라 오토마타(Celluar Automata, CA)로 발전한 후, 1980년대를 지나면서 다중 행위자 모델(Multi-agent system, M.A.S.)로 발전하였다. 콘웨이의 생명 게임(Gardner 1970), 울프람의 CA(Wolfram 1983; Wolfram 1984), 행위자 기반 모델링(agent modelling, ABM)을 통한 새 무리(bird flock)의 재현(Reynolds 1987) 등이 유명한 사례로 알려져 있다. 일반적인 시스템 시뮬레이션 방법이 시스템의 구조를 정의하고, 시스템의 정해진 구조 내부에서 매개변수가 어떻게 변화하는지를 분석하는 반면, 행위자 기반 시뮬레이션은 시스템을 구성하는 행위자의 거동 규칙만 정의하여. 행위자 집단에서 발현하는 특징을 분석한다는 점에서 차이가 있다. 다중 행위자 모델의 특징은 자율성, 지역적 관점, 탈집중화로 구분하는데, 개념적으로 GASS와 상당 부분 일치한다. 반면, 행위자 기반 시뮬레이션은 주로 행위자들의 조직적 행태에 관심이 갖는데, GASS는 각 컴포넌트의 내부 속성 변화에 관심을 두는 점에서 차이가 있다. 행위자 기반 시뮬레이션 도구로는 Repast, NetLogo, AnyLogic, SeSAm 등의 프로그램이 있다(Collier 2003; Emrich et al. 2007; Klügl et al. 2006; Tisue and Wilensky 2004). 행위자 기반 시뮬레이션 도구는 시뮬레이션을 수행하는 동안 행위자의 움직임을 실시간으로 확인할 수 있으며, 시스템 전체의 특성은 시뮬레이터에서 모니터링하여, 통계를 제시하거나 그래프로 도시하여 시스템 거동을 설명하는 형태로 이용된다는 특징이 있다.

마지막으로 시스템 다이내믹스는 1950년대 MIT 교수인 제이 포레스터 교수에 의해 창안되었다(Forrester 2007). 순환 구조(feedback loops)와 stock, flow로 설명되며, 시스템의 비선형 메커니즘을 설명하는 데 주로 이용된다. 시스템 다이내믹스는 시스템을 구성하는 인과지도를 작성하여 시스템의 구조를 확정한 후, 모니터링된 입출력 자료를 이용하여, 인과지도의 가중치를 결정하고, 가중치가 입력된 인과지도를 바탕으로 시스템의 비선형 행태를 모의한다. 대표적인 소프트웨어로 Vensim과 STELLA가 있으며(Eberlein and Peterson 1992; Richmond 1994), 시나리오를 모의하고 비교하는 연구에 강점이 있다. 시스템을

구성하는 요소의 조건의 변화에 따라 모의 결과를 저장하여, 각 시나리오별로 비교가능한 정보를 제공함으로써 의사결정자가 시스템을 이해하고. 판단을 내리는 데 도움을 준다.

#### **2.3 GASS**

#### 2.3.1 개요

Yi (2003)는 다양한 농업용 시뮬레이션 모델들의 모듈성을 향상시 킴으로써 시스템의 복잡도를 낮추고, 자기 조직 과정을 모의할 수 있는 GASS를 개발하였다. 궁극적으로 다양한 요인이 영향을 주고받는 복잡계로서, 농업 시스템에 대하여 범용적으로 활용가능한 시뮬레이션 환경을 제공하고자 하였다. 이를 위해 데이터와 메소드를 분리하고, 컴포넌트 간에 정보 전달을 일방향으로 제한함으로써 모델 결합으로 발생하는 복잡도를 낮추었다. 또한 컴포넌트를 결합하여 점진적으로 시스템을 확장하면서 모의할 수 있으며, 단위 시간을 중심으로 동시성(concurrent)을 보장하여, 개별 컴포넌트는 비동기적으로 기능을 수행하되 전체를 통괄하는 시스템 시간과 동기화함으로써 전체 시스템의 모의가 가능하도록 설계하였다.

객체지향 패러다임은 시스템을 구성하는 모든 요소를 객체로 정의하고, 각 객체를 캡슐화하여 다른 객체에서 임의로 접근할 수 없도록 제한함으로써 시스템의 복잡도를 낮추고자 하였다. 객체지향 패러다임을 차용하여, GASS는 컴포넌트라는 단위요소로 모델을 정 의한다. 더 나아가 GASS에서는 컴포넌트를 구성하는 데이터와 메 소드를 분리하여, 데이터를 데이터베이스로 관리함으로써 데이터 관 리의 효율을 높이고자 하였다.

컴포넌트를 중심으로 설계된 GASS가 다른 시뮬레이션 환경과 차별화되는 가장 큰 특징은 일방향 정보전달기법이다. 여러 종류의 컴포넌트가 존재할 때, 인접한 컴포넌트를 조사하여, 처리할 수 있 는 형태의 정보를 가지고 있을 때만 해당 정보를 취득하는 방식을 의미한다. 예를 들어, Figure 2와 같이 토양 컴포넌트(SOIL), 작물 컴포넌트(CROP), 대기 컴포넌트(ATMOSPHERE) 3가지 컴포넌트 로 구성된 작물생육 모형을 가정해 보자. CROP 컴포넌트에서는 CO2와 water 변수를 제어할 수 있고, 인접한 SOIL 컴포넌트에서 는 water 변수만 소유하므로, water 변수의 값을 읽어서 CROP 컴 포넌트를 계산한다. 반면 SOIL 컴포넌트는 관리하는 변수가 없으므 로, CROP 컴포넌트가 인접해 있음에도 아무런 작업을 수행하지 않 는다.

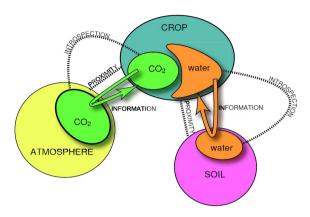


Figure 2 Concept of unidirectional information fetch scheme (Yi, 2003)

이는 PULL방식으로 명명할 수 있으며, 요청은 자유롭게 할 수 있으나, 그 응답을 보장하지 않는 시스템으로 느슨한 연결성을 보장한다. 느슨한 연결은 객체지향 패러다임의 주요한 원칙 중 하나로, 높은 응집도와 낮은 결합도를 지원하며(Gamma et al. 1994; McConnell 2004), 개별 요소의 자유도를 향상할 수 있다(Beck and Diehl 2011). 웹 기반 Open API의 근간이 되는 REST(Representational state transfer) 개념 역시 느슨하게 연결된 분산된 서비스 간의 정보 교환을 가능케 함으로써(Fielding 2000), Web 2.0 시대를 견인하였다(O'reilly 2007).

이러한 독립성을 보장하는 컴포넌트 기반의 환경을 제공함으로 써, 처음부터 시스템을 모두 정의하는 하향식(top-down) 방식이 아닌, 점진적으로 시스템을 확장하면서 개발하는 상향식(bottom-up) 방식으로 모델링이 가능하다. 시스템을 구성하는 개별 컴포넌트가 전체 시스템을 알지 못한 상태에서 단위 요소로서 주어진 임무만을 정의한 후, 인접한 컴포넌트를 인지하여 전체 시스템을 모의함으로써 연구자는 단위 요소의 거동에 집중할 수 있다. 모델을 만들 때는 전체 시스템과 무관하게 단위 모델만을 개발하면 충분하기때문에 개발에 필요한 노력이 최소화된다. 모델을 사용할 때는 개발된 모델의 연결만 정의하면 주어진 도메인에 대하여 모의할 수 있기 때문에, 기존 시뮬레이션 환경에서 여러 모델을 결합하기 위하여

메인 함수에서 절차와 자료 교환 과정을 정의하는 방식에 비하여 간편하게 이용할 수 있다. 이러한 특징은 관개 시스템 모의, 열전도 모의, 농산물 유통 모의와 같은 다양한 분야의 문제해결에 적용되어 검증되었다(Yi 2003; 김태곤 2007; 서교 et al. 2005).

분석 대상 모델이 되는 단위 컴포넌트는 매 단위시간마다 인접한 컴포넌트의 값을 이용하여 자신의 컴포넌트 속성을 업데이트하며, 이 때 인접한 컴포넌트에 인위적으로 값을 주입할 수 없다. 이는 방어적 프로그래밍 개념으로, 시스템을 구성하는 객체가 서로 레퍼런스를 공유하면서 값이 의도치 않게 전파되면서 발생하는 부작용을 방지하는 역할을 수행하여 시스템의 안전도를 향상시킨다(Bloch 2008; Goodliffe 2007). 방어적 프로그래밍 개념을 기반으로 각 컴포넌트는 동시에 수행될 수 있었으며(Peierls et al. 2006), 일정 시간마다 동기화를 통하여 각 컴포넌트의 정합성을 유지하였다.

#### 2.3.2 인접성을 이용한 연결

기존 GASS는 시스템을 구성하는 구성요소간의 연결정보를 컴포넌 트의 위치를 이용한 인접 여부를 기준으로 판단한다. 다음 Figure 3 과 같이 컴포넌트 A, B, C가 존재할 때, 시뮬레이션 환경 GASPanel 위에 각 컴포넌트의 인스턴스를 올려둠으로써 시뮬레이션을 구동할 수 있다. 이 때 컴포넌트의 인스턴스가 서로 가까이 위치한다면, 두 인스턴스는 정보를 주고 받을 수 있게 되며, 그렇지 않은 경우에는 정보 교환 없이 독립적으로 컴포넌트의 기능을 수행한다. 이와 같은 인접성을 이용한 연결정보를 정의함으로써, 사용자가 시각적으로 관계를 파악하고, 단순히 위치를 이동시키는 작업을 통하여 간단하게 관계를 정의할 수 있다.

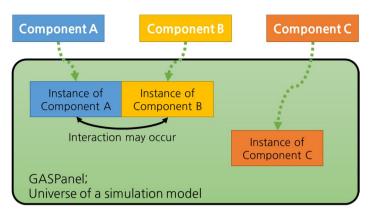


Figure 3 Schematic view of established simulation model (Yi. 2003)

이와 같이 GASS에서 컴포넌트의 연결을 물리구조적 인접성을 기준으로 설정한 까닭은 시스템을 구성하는 요소가 MPC 모형에 기반한다고 가정하였기 때문이다. MPC 모형은 Multilayered Primitive—Composite Models의 약자로, Primitive—Composite 모형을 개량한 모형이다(배연정 2001). 시설물을 여러 단계로 구분하기 위한 모형으로, 시설 레이어, 부분 레이어, 요소 레이어, 기본형레이어로 구분되어, 각 레이어별로 특성을 계층적으로 통합하여 전체 시설물을 설명할 수 있다. 이러한 실제 시설물을 바탕으로 구성된 분류법에 따라 시스템을 분해하였기 때문에, 시스템을 구성하는요소, 즉 컴포넌트의 인스턴스는 실존적 공간을 가진다고 가정하였다. 따라서, 농업 기반 시설물을 분류하고 모사하는 데에는 적합한모형이라고 할 수 있다.

### 2.3.3 속성정보 관리

GASS는 속성정보 관리를 위하여 JAZZ 프레임워크를 이용하였다. JAZZ 프레임워크는 서울대학교 농업구조 및 시스템 연구실에서 개발된 통합 데이터베이스 프레임워크로, 다양한 벤더의 데이터베이스 시스템을 동일한 환경에서 사용 가능하도록 제공하였다. 그러나 수년의 시간이 흐르는 동안 벤더의 기능과 프로토콜이 바뀌는 데 반해 JAZZ 프레임워크는 지속적으로 업데이트를 할 수 없었고, 현재에 이르러서는 JAZZ 프레임워크를 활용할 수가 없게 되었다.

현재에 이르러 원활하게 사용할 수 없음에도 불구하고, GASS

는 데이터베이스를 이용하여 속성정보를 저장할 수 있었으며, 속성 정보는 JazzBean 기반의 컴포넌트 기술을 이용하여 각 클래스의 필드로 저장하고, 해당 필드의 접근자를 통해 정보를 주고 받을 수 있도록 개발되었다. JazzBean 기술은 java에서 제안한 기술로 특정 한 규약을 만족할 경우, JazzBean Editor를 이용하여 손쉽게 데이터 객체를 만들고, 제어할 수 있는 기술이었다(Englander 1997).

컴포넌트 간의 정보 전달은 각 컴포넌트에서 원하는 정보에 해당하는 키워드를 이용하여 인접한 컴포넌트에 자료를 요청한다. 요청받은 컴포넌트에 해당 키워드의 변수가 있을 경우, 값을 반환하지만 그렇지 않은 경우에는 아무런 작업을 수행하지 않음으로써, 다양한 컴포넌트가 결합된 시스템에서 원활하게 시뮬레이션이 구동될수 있도록 설계하였다.

## 2.4 시스템 시뮬레이션 도구의 한계

시스템 시뮬레이션 도구의 기능을 검토하기 위하여 연구대상이 되는 지역시스템의 관심문제를 정의하고자 하였다. 지역시스템은 자연현상과 이에 대응하기 위한 인간의 노력을 분석하려는 학문으로, 이에 해당하는 문제는 이질적인 요소들이 결합되고, 대부분 연속된 공간 속에서 상호작용을 하지만, 상호작용의 속도가 일정한 단위를 갖지 않는 형태의 문제로 정의할 수 있다.

대표적인 시스템 시뮬레이션 도구인 LabView와 Simulink는 시그널 처리, 행렬 계산, 최적화, 그래프와 같은 다양한 유틸리티를 바탕으로 사용자층이 가장 두터운 응용프로그램이다. 전기회로를 모티브로 시작된 모형으로 다양한 분야에 응용되고 있으며, 기계장치의 반복적 행태를 모사하는 데 매우 유용하다. 그러나 앞서 정의한 지역시스템 분야의 문제를 모의하기에는 기술적으로 가능하지만, 효과적이지 못하다. 범용적으로 사용될 수 있는 반면, 지역시스템 분야의 문제를 풀이하기 위해서는 간단한 모형도 복잡한 수식과 관계를통해서 표현되어야 하는 경우가 발견된다.

시스템 다이내믹스 도구는 stock과 flow를 이용하여 시스템을 구성하고, 피드백 구조를 통하여 시스템을 재현하는데, 이는 인공신 경망 모형에서 신경망 구성과 가중치와 유사한 개념이다. 다만 시스 템 다이내믹스에서는 연결구조를 전문가의 의견 교류를 통하여 인 과지도라는 유의미한 형상을 구성하는 데 반해, 인공신경망 모형에 서는 기계적으로 무의미한 형상을 구성하여 시스템을 가장 잘 모사하는 연결구조를 거꾸로 찾아가는 방식으로 모델을 발전시킨다. 이러한 시뮬레이션 도구는 논리적 관계에 기반을 두고 있기 때문에, 형상을 해석하는 문제에 취약하다. 그리드 자료를 활용한 모형이나유한요소법, 유한차분법, 유한체적법과 같이 유한공간의 적분을 이용하는 모델을 재현하는 데 적절하지 않다.

행위자 기반 모델링 중심의 시스템 시뮬레이션 도구는 구성요소들의 간단한 규칙을 통하여 거시적 동태가 발현할 수 있음을 모의실험을 통하여 관찰한다. 전염병의 확산과 같이 개개인의 메커니즘은 단순하게 감염으로 정의하는 반면, 개인별로 잠복기와 감염기간의 차이가 있고, 개개인의 접촉빈도에 따라 감염확률이 달라지는문제에 대하여 전체 집단의 감염율을 쉽게 모의할 수 있고, 이를 이용하여 공동체 공간의 격리시기를 결정할 수 있다. 이러한 예제와같이 행위자 기반 모델링은 구성요소간의 관계를 정의하지 않고, 무작위적인 관계에서 새로운 규칙이 발현하는 과정을 관찰한다. 그러나 지역시스템 분야의 문제는 공학적 지식을 바탕으로 관계가 규명된 모델이 상당 부분 존재하며, 관리법, 규정과 같이 일정하게 정해진 모형의 운영규칙을 시뮬레이션 환경에서 반영하고자 하는 요구가 있다. 행위자 기반 모델링 기반의 시뮬레이션 도구는 이러한 요구사항에 효율적으로 대응하기 어렵다.

이상에서 살펴본 바와 같이 대표적인 시스템 시뮬레이션 도구들에 대하여 지역시스템 분야의 적용가능성을 평가하였다. 그러나 사례별로 시스템 시뮬레이션 도구의 장단을 파악하는 데는 한계가 있으므로, 도구의 근간이 되는 메커니즘을 중심으로 도구를 평가할 필요가 있다. 가장 이상적인 모습에 가까운 시스템은 매우 많은 수의 구성요소가 결합하여, 생명유지라는 공통의 목적을 성공적으로 수행하는 생명체라고 정의할 수 있다. 이러한 생명체를 모의할 수 있는 시뮬레이션 도구라면, 비교적 단순한 지역시스템도 모의가 가능할 것으로 판단하였다.

생명체는 수많은 세포가 유기적으로 결합하여 생명을 유지하는 대표적인 시스템이기 때문에, 많은 연구가 생물의 내부 메커니즘에서 영감을 얻고, 이를 모사하려고 노력한다. 콘웨이는 생물의 활동을 탄생, 생존, 죽음으로 구분하여, 생명 게임(Conway's game of life)을 소개하며, 셀룰라 오토마타라는 분야를 창안하였으며

(Gardner 1970), 이후 많은 복잡계 연구가 이러한 개념을 기반으로 발전하였다. 농업 시스템을 모의하기 위한 목적으로 개발한 GASS 또한, 농업 시스템의 근간이라고 판단되는 식물의 생장 메커 니즘을 통하여 모델을 설명하였다.

생명체는 신진대사를 통해 존재목적인 생명을 유지하고, 자기복제를 통하여 증식하며, 증식 과정에서 형질을 변화하는 기능을 수행하는 대상으로 정의할 수 있다(루흐알라 2012). 신진대사는 각 세포에서 발생하는 생화학적 반응으로, 독립적으로 작용하기도 하지만, 주변 세포와 물질교환과 정보교환을 통하여 발현한다. 물질교환은 인접한 세포 간 융합, 삼투 현상 등으로 설명되며, 정보교환은 신경계에서 신경세포를 통하여 전기 에너지 형태로 정보가 빠르게 전달되는 현상으로 설명되다.

Table 1 Comparison of GASS components and cells

	GASS (GASS	Biology (Cells)
	components)	
Connection	Proximity	Proximity
		Link
Communication	Homogeneous	Homogeneous
		Heterogeneous
Stages	Active	Active (Survival)
		Birth
		Death

GASS는 식물을 모티브로, 잎에서의 광합성 및 뿌리에서의 영양물질교환작용을 모사하여 독립적으로 수행 가능한 컴포넌트를 설계하였다. 시스템을 구성하는 개별 요소들이 독립적으로 수행되며, 시스템을 확장하여 다른 요소들과 연결시킴으로써 구성요소 간의정보교환을 통한 시스템의 자기조직화를 모의하고, 복잡한 시스템의창발을 확인하였다. 그러나 다수의 요소들이 결합되면서 요소들 간의 속도가 다른 정보가 존재할 수 있고(Han et al. 2012), 이들 정보를 효과적으로 표현하기 위해서는 정보전달에 있어 정보계와 물질계가 분리될 필요가 있다고 지적된 바 있다(김태곤 and 이정재2007; 이형진 2014). GASS의 근본적인 한계를 찾고 개선하기 위하여, 개념적 근거가 되었던 생명체와 비교하였다. 생명체를 구성하

는 세포와 GASS의 구성요소인 컴포넌트의 특징을 비교하여 Table 1과 같이 정리하였다.

세포는 인접한 세포와 물질교환을 할 뿐만 아니라, 신경세포를 통하여 빠르게 정보를 전달받기도 하는데 반해 GASS에서는 인접한 컴포넌트와의 정보전달만 가능하여 정보전달에 취약한 문제가 있다. 이에 정보계와 물질계를 분리하여, 정보전달을 효과적으로 할 수 있는 방법으로 전체 시스템에 브로드캐스팅 방식이 제안되었다(김태곤 and 이정재 2007). 그러나 생명체는 신경세포를 이용하여 신경계에서 정보계를 관장하는 것에 비해 브로드캐스팅 방식은 수신세포를 제한할 수 없다는 점에서 한계를 갖는다. 이에 정보계와 물질계가 분리될 필요가 있지만, 정보계를 구현하는 데 있어서 특성 요소에만 정보를 전달할 수 있는 위상구조의 도입이 필요하다(한이철 2012).

세포의 신진대사를 살펴보면, 같은 기관이나 조직 내에서 작용하는 행태와 다른 기관에 속한 조직 간 작용이 다르다. 같은 조직내에서는 서로가 독립적으로 작용할 수 있는 반면, 다른 조직 간에는 프로토콜이 맞아야만 작용을 할 수 있다. 이러한 현상은 같은 조직 내에서 서로 독립적인 위상으로 동일한 형태로 통신할 수 있는 동질소통구조와 어느 그룹에 속한 세포인지를 확인하여 그룹에 따라 다른 형태로 통신하는 이질소통구조로 정의할 수 있다. 동질소통구조는 전후 상태와 상관없이 연결되어 있는 함수만으로 설명할 수 있는 구조인 반면, 이질소통구조는 동일 위상에 연결되어 있는 작업들의 결과에 따라 현재 작업이 달라지므로, 동일 위상에 존재하는 함수, 즉 동일 그룹의 함수에 따라 결과가 달라지는 구조로 설명할수 있다. 이러한 측면에서 기존 GASS 환경은 동질소통만을 고려하여, 그룹화된 문제를 해석하는 데 한계가 있다.

마지막으로 시스템의 구성이 변화하는 문제, 즉 시스템이 확장되거나 축소되는 모형에서는 GASS를 이용하여 모사하기가 어려웠다. 세포는 자기 복제를 통한 생성, 세포 성장이나 신진대사와 같은 활동작용, 최종적으로 사멸에 이르는 단계를 거치게 된다. 셀룰러오토마타, 행위자 기반 시뮬레이션 기법은 이러한 세포의 생명주기를 모티브로 발전하여 복잡계 문제들을 재현하였다(Macal and North 2010; Santl et al. 2010). 그러나 GASS는 시스템의 구조가고정되어 있는 문제에 한해서만 모의할 수 있다는 한계를 지니고

있다.

이상에서 지적한 바와 같이 시스템 시뮬레이션 도구가 갖는 3 가지 측면의 한계를 인식하고, 이를 개선한 농업시스템응용플랫폼 (ASAP)을 개발하였다. 기존의 GASS의 장점인 일방향 정보전달기법, 시각 프로그래밍, 속성정보 관리기법 및 독립성이 유지되는 컴포넌트 기반 모델링 등은 계승하고, 지적된 한계를 개선함으로써 지역시스템 기반 시뮬레이션 모델의 재현성을 높이고자 하였다.

## 제 3 장 ASAP의 개발

#### 3.1 발전된 시뮬레이션 프로토콜

#### 3.1.1 연결 프로토콜

GASS는 컴포넌트의 연결성을 설정하기 위하여 물리구조적으로 인접한 컴포넌트를 검색하는 방식으로 구현되었다. 이는 현실세계에서 물리적 공간을 점유하는 실존적 대상에 대하여 모델링을수행한다는 개념에서 출발했기 때문이다. 모델링으로 활용되었던대상을 살펴보면, 관개시스템 모의에서는 논, 수로, 유역, 저수지,작물(벼), 대기를 대상 컴포넌트로 정의하였으며, 구조시스템에서는수문과 강, 바다로 정의하였고, 또 다른 구조시스템에서는물, 사면,포화토, 불포화토로 정의하였다. 이들 컴포넌트는 연결성이단순하여, 2차원 공간상에서 인접하게 위치시킬 수 있었기 때문에효과적으로 시스템 시뮬레이션이 가능했다. 그리고 농산물 유통을모의하기 위하여 추상적인 모델인 생산자, 도매상인, 소매상인으로컴포넌트를 구성하더라도(서교 et al. 2005), 복잡하지 않은 관계는물리구조적 인접성에 기반한 컴포넌트 연결이 가능하였다.

그러나 저수지, 수로, 논으로 구성된 관개시스템에서 다수의 수로 컴포넌트가 연결되고, 양 끝단에 위치하는 저수지 컴포넌트와 논 컴포넌트가 정보를 교환해야 하는 상황에서, 다수의 수로컴포넌트를 지나 정보가 전달됨에 따라 정보전달이 지연되고, 이로 인하여 관개시스템의 모의가 효과적으로 이루어지지 않는 한계를 지적한 바 있다(김태곤 and 이정재 2007). 전달 속도가 다른 정보에 대하여 정보계와 물질계로 분리하여 정보를 다루어야 한다고 제안하면서, 브로딩캐스트 기법에 기반한 해결방안을 제안한 바 있으나, 본 연구에서는 근본적인 대안으로 컴포넌트 간 관계를 위상구조로 연결관계로 정의하였다. 위상구조는 물리구조적으로 인접한 컴포넌트를 전처리에서 계산하여 변환함으로써, 물리구조와 위상구조를 모두 담을 수 있다는 장점이 있다.

이러한 위상구조에서의 관계된 컴포넌트간의 연결은 그래프 이론에 따라 인접행렬로 정의할 수 있다. 공학 문제는 대부분 실세계에 실존하는 문제로서 물리구조적 관계에 기반하기 때문에 인접행렬로 관계를 서술할 경우에 희소행렬일 가능성이 높다. 대표적으로 유한요소법에서 전체 강성행렬(global stiffness matrix)은 대부분 희소행렬이다(Bro-Nielsen 1998; Williams 1973). 따라서 효율적으로 인접행렬을 관리하기 위하여 리스트 구조를 이용하여 컴포넌트 관계를 관리하도록 시뮬레이터를 개선하였다.

네트워크로 정의된 전체 컴포넌트의 관계망에서 각 컴포넌트는 단계별로 연결된 컴포넌트를 찾아야 하므로, 시뮬레이션을 수행하기전에 너비우선탐색(Breadth First Search, BFS) 알고리즘을 이용하여 각 컴포넌트가 최상위 노드로 설정되는 신장 트리로 변환하였다(Cormen 2009). 이러한 변환과정을 통하여 각 컴포넌트는 모든 컴포넌트와 몇 단계로 연결되어 있는지 알 수 있으며, 일반적으로 직접 연결된 컴포넌트와의 관계, 즉 1단계연결만으로 시스템의 상호작용을 모의할 수 있다.

#### 3.1.2 소통 프로토콜

시스템 시뮬레이션은 다수의 동일한 컴포넌트가 연결되어, 서로 정보를 주고 받으며 유기적으로 기능을 수행하는 행태를 모의한다. 이 때 정보 통신 방법은 크게 동질소통구조와 이질소통구조로 나눌수 있다. 동질소통구조는 동일한 컴포넌트끼리 정보를 주고 받을 때한 가지 경로로만 정보를 주고 받는 행태로서 Figure 4와 같이도시할 수 있다. 동질소통구조를 갖는 시스템에서는 컴포넌트 간의연결 유무만 의미가 있으며, 계산을 위한 정보는 컴포넌트가 갖고있는 속성으로 모의된다. 균일한 시스템을 다루는 문제가 주로 동질소통구조로 구성된다.

반면 이질소통구조는 동일한 컴포넌트 사이에 정보 교환경로가 여러 가지 형태인 시스템에서 나타나는 구조로, Figure 5와같이 도시할 수 있다. 동질소통구조와 달리 이질소통구조는 정보전달 경로가 2가지 이상을 포함하는 시스템 해석 문제로써, 정보를전달할 때 동일한 클래스의 컴포넌트라고 하더라도, 대상개체(instance)에 따라 전달 체계가 달라진다.

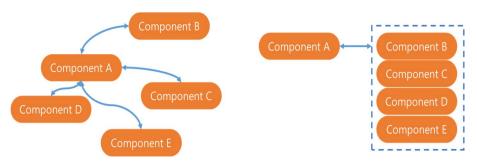


Figure 4 Concept diagram of homogeneous structure for communication



Figure 5 Concept diagram of heterogeneous structure for communication

이질소통구조를 반영하여 정보전달을 구현하기 위하여, 컴포넌트 속성 정보 중 이질소통구조에 해당하는 속성은 연결속성을 결합하여 저장하도록 구현하였다. 컴포넌트 A에서 height와 이질소통구조로 연결된 weight라는 변수가 있을 때, height는 일반변수로 저장하여 관리하였다. 반면 weight 변수는 3번 컴포넌트와 5번 컴포넌트와 연결되어 있다고 가정할 때, (weight, 3), (weight, 5)와 같이 벡터변수로 저장하여 속성을 관리하도록 개발하였다.

#### 3.1.3 복제 프로토콜

시스템이 성장하거나, 쇠퇴하는 시스템 외연이 변화하는 문제를 모의하기 위해서는 시스템을 구성하는 요소의 복제 기능이 필요하다. 요소의 복제는 단순히 동일한 인스턴스를 생성하는 데 그치지 않고. 기존 시스템 요소와의 관계를 정의해야 한다. 본 연구에서 복제를 구현하기 위하여, 복제의 권한은 각 컴포넌트가 갖는 반면, 복제를 실제로 수행하는 주체는 전체시스템을 관장하는 ASAPEnvironment 클래스가 담당하도록설계하였다. 이러한 설계는 중앙에서 복제를 결정하게 되면, 복제기능이 한 곳에 집중되어 복잡도가 높아지는 기존 시스템의 문제를답습하게 되므로, 복제의 권한은 각 컴포넌트가 갖도록 설계하였다. 반면 실제 복제 기능은 시뮬레이션의 중심기능을 수행하는시뮬레이터의 중심 클래스, ASAPEnvironment 클래스에서 관련정보를 제어하기 때문에, 각 컴포넌트에서 복제 기능이 요청되면,이를 정해진 루틴에 의해 수행하도록 개발하였다.

개념적으로 cloning 루틴은 복제를 요청한 인스턴스와 동일한 인스턴스를 생성하고, 인스턴스의 모든 정보를 복사한 후, 요청한 인스턴스에 연결된 인스턴스로 설정하여 시뮬레이션 환경에 반영하는 과정으로 이루어진다. 새로운 인스턴스는 복제된 이후, 처음에는 복제를 명령한 인스턴스와 유일하게 연결되며, 시간이 지남에 따라 시뮬레이션 환경에서 정의한 연결 규칙에 따라 새로운 연결을 확장하는 방식으로 구현하였다.

본 연구에서 작성한 cloning 루틴은 객체지향 프로그래밍 언어인 java에서 제공하는 clone 메소드 구현 원칙을 참조하여 구현하였다. Bloch (2008)는 객체를 복제하는 과정에서 mutable 객체는 주의 깊게 값을 옮겨야 하며, 깊은 복제(deep copy)가 필요하다고 역설하였다. 각 컴포넌트는 자료 저장을 위해서 사전형 자료구조인 Map 클래스와 리스트 자료구조인 List 클래스를 포함하고 있고, 이들 클래스는 mutable 객체를 생성하므로, 얕은 복제(shallow copy)가 일어나지 않도록 개발하였다. 또한 시뮬레이션 결과를 데이터베이스에 저장하기 위한 주 키(primary key)가 되는 컴포넌트의 id가 중복되지 않도록 구현하였다. 이러한 작업은 일정한 규칙에 따라 반복되는 작업이면서, 동시에 규칙을 지키지 않거나, 사용자의 실수에 의해 시스템이 오작동할 수 있는 여지가 있다. 따라서, ASAP 시뮬레이션 환경에서 복제 기능을 제공하여 사용자가 코딩을 하지 않고 이용할 수 있도록 편의성을 제공하고자 하였다. 이러한 cloning 기능을 활용하여 사용자는 시스템에서 제공하는 동일한 컴포넌트를 손쉽게 생성할 수 있다.

## 3.2 시뮬레이터의 설계

#### 3.2.1 컴포넌트 개발 프로토콜

시뮬레이션을 관장하는 ASAPEnvironment 클래스에서는 Figure 6에 도시한 바와 같이, IASAPComponent 인터페이스를 상속한 인스턴스를 소유하고, simulation() 메소드가 호출되면, 단위시간(tick)을 증가시키면서 각 컴포넌트가 preProcess(), process(), postProcess() 메소드를 수행되도록 관장한다. 특히 process() 메소드는 앞서 각 컴포넌트 별로 인접한 컴포넌트를 전달함으로써, 컴포넌트 간에 정보전달이 가능토록 설계하였다.

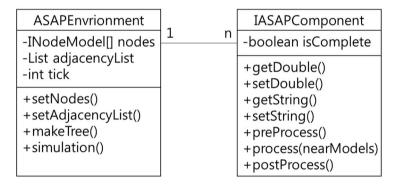


Figure 6 Class diagram of main class and component interface

컴포넌트의 프로토콜 역할을 수행하는 IASAPComponent 인터페이스는 각 컴포넌트의 종료여부를 판단할 수 있는 isComplete 필드를 가지고 있으며, 실수형과 문자형 정보를 저장, 열람할 수 있도록 접근메소드(getter/setter)을 가지고 있다. 연구자는 IASAPComponent를 구현한 DefaultModel라는 추상 클래스를 상속하여 프로그램을 작성함으로써 모델 구현 과정을 간소화할 수 있으며, 주로 process() 메소드만 작성하면 모형을 구현할 수 있도록 설계하였다.

기존 GASS에서는 컴포넌트의 명시적인 규약이 부족하고, java 언어의 리플렉션(reflection) 기능을 활용하여, 바이트코드에서 메타정보를 찾는 방식(introspection)으로 구현되었다. 이러한 방식으로 인하여 설계 상의 복잡함이 존재하고, 개발된 컴포넌트 사이에 일관된 형태를 갖추지 못하였다. ASAP에서는 컴포넌트의 규약을 인터페이스를 이용하여 구현함으로써, 설계를 간결하게 변경하고, 컴포넌트 개발을 위한 코드 작성 시에도 더 적은 코드로 일관된 형태로 작성할 수 있도록 하였다. 이를 통해, 새로운 사용자가 ASAP를 통해 모델을 작성하는 방법을 익히는 데 필요한 노력이 줄여들 것으로 예상된다.

#### 3.2.2 컴포넌트 이용 프로토콜

컴포넌트를 이용하는 연구자는 시스템 시뮬레이션을 수행하기 위하여, 다음과 같은 프로시저로 프로그램을 작성해야 한다. 먼저 사용할 컴포넌트의 인스턴스를 생성하고, 각 인스턴스에 속성값을 부여한다. 인스턴스의 속성은 분석 모델 구현 시 필요한 변수들과 GASS에서 특수하게 사용되는 키워드로 구분된다. 시스템에서 제공하는 키워드는 Table 2에 정리하였다. 이들 키워드를 통하여 시뮬레이션 구동 과정에서 컴포넌트를 시각화하기 위한 색상, 크기, 모양, 위치 등을 제어할 수 있다. 특정 컴포넌트의 속성을 관찰해야할 때에는 "plot" 키워드를 이용하여 그래프로 값을 추적할 수 있다. 또한 시각화된 컴포넌트의 속성은 시간 흐름에 따라 도메인이 변화되는 정성적 정보로 활용될 수 있으므로, 시뮬레이션 구동 이미지를 저장할 수 있는 자동 이미지 저장 기능을 제공한다.

Table 2 Reserved words for ASAP

Keywords	Meaning	Type (example)
х	Position x in canvas	Double (500, left side is zero)
У	Position y in canvas	Double (300, bottom is zero)
size	Size ratio of symbol	Double (default: 1)
color	Color of symbol	Color name or RGB value
		("red", "blue", "#6699FF")
shape	Shape of component symbol	Shape name or filename
		("rect", "circle", "or.png")
plot_item	Item name to plot chart	String ("height")
plot_list	Values for scatter chart	Double list (1.5, 2, 3.5, 5)
capture	Tick list for snapshot	Integer list (10, 20, 30)

개별 컴포넌트 인스턴스에 대한 설정이 끝나면, 컴포넌트를 연결하는 인접리스트를 정의한다. 일반적으로 방향이 없는 시스템의 경우에는 한쪽 방향만 지정하면, 반대방향의 연결도 자동으로 설정된다. 컴포넌트의 연결정보는 시뮬레이션 구동시 화면상에서 확인 가능하므로, 누락된 링크가 없도록 주의가 필요하다.

컴포넌트의 연결성을 정의하면, 기본적으로 시뮬레이션 구동을 위한 설정은 완료된다. 추가적으로 시뮬레이션 구동화면을 저장할 수 있도록 특정 시점을 정의할 수 있고, 시뮬레이션 데이터를 데이터베이스에 저장한 후, csv(comma-separated values) 파일로 추출할 수 있다.

#### 3.2.3 시뮬레이션 유틸리티

본 연구에서는 시스템 시뮬레이터를 개선하면서, 연구에 필요한다양한 편의기능을 추가하였다. 먼저, 시뮬레이션 결과를 저장하기위하여 경량 데이터베이스를 도입하였다. 경량 데이터베이스는데이터베이스 관리시스템 구동을 위하여 별다른 프로그램 설치없이 기존 프로그램에 접착 프로그램(glue program) 형태로제공할 수 있다는 점에서 배포의 장점을 갖는다. ASAP으로시뮬레이션된 결과는 경량 데이터베이스인 SQLite를 통하여데이터베이스 파일로 저장되며, 시뮬레이션이 종료된 후, csv파일로 변환하여, 엑셀로 쉽게 원하는 데이터를 참조할 수 있도록유틸리티를 제공하였다.

두 번째로, 실시간으로 시뮬레이션 결과를 확인할 수 있도록 그래프 기능을 추가하였다. 그래프는 JFreeChart 라이브러리를 이용하여 구현하였으며, "plot\_item"으로 지정한 속성에 대하여 시뮬레이션이 수행되는 동안 값을 추적하여 결과를 그래프로 시각화하도록 개발하였다. 그래프 기능은 주요 관심 영역의 속성 값변화 및 시스템 시뮬레이션을 수행하는 동안 오차의 수렴을 관찰하는 데 이용된다.

세 번째로, 시뮬레이션이 구동되는 동안 시뮬레이션 모델의 구성을 파악할 수 있도록 네트워크 뷰어를 개발하였다. 네트워크 부어는 컴포넌트의 연결상태를 보여주며, 컴포넌트에서 지정한 색채 및 형태로 표시하게 되는데, 특정 속성값에 따라 색채를 변경하면, 시뮬레이션 진행 상황을 정성적으로 쉽게 확인할 수 있다. 확산 문제와 같은 유한차분 해석방법에서는 후처리 없이 바로 결과를 시각화한다는 점에서 장점이 있다.

이러한 유틸리티는 GASS에서 제공하지 않거나, 제공되더라도 편의성을 향상시킬 필요가 발생하여 개발하였다. 데이터베이스는 기존 GASS에서 jazz 프레임워크를 이용하여, 널리 이용되는 RDBMS(Relational database management system) 중 하나인 MySQL를 사용하였다. 서버 환경에서 데이터베이스를 서비스하는 경우에는 이용하는 데 어려움이 없으나, 로컬 환경에서 데이터베이스를 이용하는 경우에는 따로 MySQL 서버를 구동하고, GASS에서 이용할 수 있도록 설계된 스키마에 따라 데이터베이스를 정의해야 했다. 이러한 환경은 GASS를 원활하게 배포하기 위해서 데이터베이스 서버를 운영해야 제약이 생기게 되었다. 반면, 본연구에서는 경량 데이터베이스를 이용함으로써, 다소간의성능저하가 발생하지만 배포에 유리한 설계를 이루어냈다. 자료 저장 과정에서 배치 처리를 통하여 성능을 개선하여, 데이터베이스 교환에 따른 부작용을 최소화하였다.

GASS 환경에서는 결과를 확인하기 위하여, 시뮬레이션 구동후 데이터베이스에 축적된 자료를 쿼리를 통해 추출하여, 엑셀과 같은 프로그램을 이용하여 비교하였다. 반복적으로 결과를 확인해야하는 상황에서는 매우 번거로운 과정이었기 때문에, ASAP에서는 그래프와 네트워크 뷰어를 통하여 결과를 바로 확인할 수 있도록 제공하였다. 결과를 상세히 분석하기 위해서는 데이터베이스에 저장된 결과값을 추출하여 후처리(post process)하는 과정이 필요하지만, 결과값의 경향이나 분포를 실시간으로 제공함으로써, 사용자가 즉각적으로 모델을 수정할 수 있도록 하였다. 이를 통해, 연구자는 대화형으로 시뮬레이션 모델을 다룰 수 있다. 이러한환경을 통해 ASAP가 제공하고자 하는 가치는 애자일 선언(Beck et al. 2001)에서 성공적인 소프트웨어 개발을 위해 제시한 가치와 맥을 같이 하고 있다.

#### 3.3 연결성에 대한 검토

#### 3.3.1 물리구조로 연결된 시스템

물리구조로 연결된 시스템은 일정한 공간을 가지고 연속적으로 존재하기 때문에, 기하학적으로 표현하기에 용이하다. 일반적으로 공학적 해석을 위하여 모델링을 수행하는 과정에서 현실을 있는 그대로 모사할 수 없고, 추상화를 통하여 일부 특징을 중심으로 관계를 기술하는데 반해, 물리구조로 연결된 시스템은 상대적으로 덜 추상화되어, 현실에 존재하는 구체적 형상에 가깝게 사상하여 모델링한시스템이다. 물리구조로 연결된 시스템은 기존 GASS에서 제시한방법과 같이, 인접성을 기반으로 요소들의 관계망을 정의할 수 있다.

대표적으로 주어진 경계 내에서의 확산 문제는 잘 알려진 물리 구조로 이루어진 시스템이다. 확산 문제는 평형상태로 도달하려는 자연의 법칙을 수식화한 문제로 일반적으로 식 (1)과 같이 Laplace equation으로 정의하며, 유한차분법(Finite Difference Method, FDM)과 같은 수치해석 방법을 이용하여 해석한다.

$$\nabla^2 u = 0 \tag{1}$$

본 연구에서는 확산 문제 중 하나로 2차원 열전도 문제를 선정하고, Figure 7과 같이 구성하였다. 주어진 경계조건은 좌측 경계는 100℃이며, 나머지 경계는 0℃로 정의되며, 문제의 해는 내부 온도가 평형상태에 이르렀을 때의 온도를 의미한다. 이러한 열전도 해석문제는 컴포넌트가 간격이 일정한 2차원 격자 형태로 연결되어 있어서 각 컴포넌트가 해당 격자의 값을 대표하여 서로 일 대 일로대치가 가능한 물리구조를 갖는 시스템이다.

또한 열에너지는 방향성이 없고 높고, 낮은 위상 차에 의해서만 움직이므로 동질소통구조를 갖는 시스템이다. 이 문제는 GASS를 이용하여 모의한 예제로서(김태곤 2007), 물리적으로 연결된 구조와 동질소통만으로 이루어져 있어 GASS로 풀이가 가능하였다. 주어진 경계조건에서 내부 11개 지점이 정상상태에 도달하였을 때의온도를 모의하기 위하여 ASAP를 이용하여 시뮬레이션을 실시하였다.

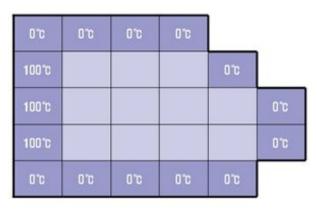
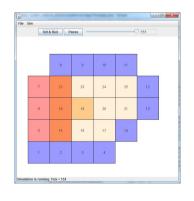
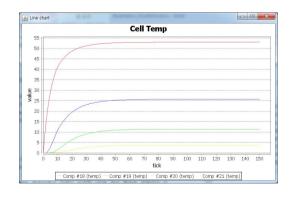


Figure 7 Problem of 2D heat conduction (Kim. 2007)

제시한 문제를 풀이하기 위하여 열전도 컴포넌트를 구현하고, 컴포넌트를 연결하여 시뮬레이션 환경을 정의하였다. 열전도 컴포넌트는 정상상태를 목표로 작동하므로, 변화율의 변화가 없는 상태로 정의할 수 있으며, 달리 표현하면 각 컴포넌트의 속성값은 주변 컴포넌트가 갖는 속성 값의 평균으로 정의할 수 있다. 이와같은 규칙으로 작성된 컴포넌트를 서로 연결하여 주어진 문제도메인과 같은 형태로 시뮬레이션 환경을 구축하여 ASAP를실행시킨 결과, Figure 8에 도시한 바와 같이 두 가지 결과물을확인할 수 있다. 왼쪽 창은 주어진 도메인에서 실시간으로 온도에따라 색이 변하는 행태를 실시간으로 그려주며, 오른쪽 창은 내부지점 중 2행의 4개 열에 해당하는 지점의 온도변화를 서로 비교가능하도록 그래프로 결과를 제공한다.





### 3.3.2 위상구조로 연결된 시스템

위상구조로 연결된 시스템은 공간적인 위치보다는 구성요소 간의 관계를 중심으로 설명되는 시스템이다. 공정관리를 위해서 사용되는 PERT(Program evaluation and review technique)는 여러 작업 공정이 네트워크로 연결된 시스템으로서, 프로젝트를 실행하는 데 있어서 단위작업을 요소로 정의하고, 각 요소의 예상시간과 선행, 후행 요소를 할당하여, 전체 프로젝트의 예상 완료시간을 추정한다. 즉, 현실 세계에 존재하는 유형의 물체를 모델링한 것이 아니라, 추상적인 개념인 작업 공정시간을 모델링한 시스템이다. 이와 같이 시스템을 구성하는 단위요소의 연결 여부를 정의하는 데 있어서 거리 개념을 적용할 수 없는 네트워크 기반의 모델은 대표적인 위상구조를 갖는 시스템이다.

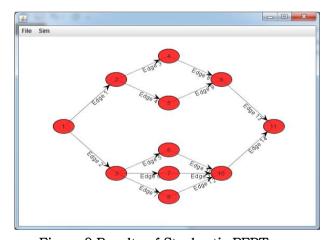


Figure 9 Results of Stochastic PERT

확률론적 PERT는 PERT와 달리 임계 경로를 단일하게 한정하지 않고, 각 경로 별로 임계 경로가 발현할 확률을 모사한 기법이다(윤득노 et al. 2010). 본 연구에서는 윤득노 et al. (2010)가 제시한 컨볼루션, PRO(parallel reduction operator), PCPC(Probabilistic critical path calculator)를 이용하여 컴포넌트를 개발하고, 이를 이용하여 Figure 9와 같이 확률론적 PERT를 구현하였다. 동질소통구조로 연결된 위상구조 문제로서,

개발된 컴포넌트를 이용하여 손쉽게 구현할 수 있었으며, 모의 결과는 발표된 연구결과와 동일한 결과를 확인하였다.

### 3.3.3 혼합 시스템

앞서 물리구조를 갖는 대표적인 문제로 그리드 형태로 구성된 2차원 열전도 문제를 예로 제시하였다. 열전도 문제에서 특정 지점의 온도에 따라 외부 경계 조건을 달리하는 컨트롤러가 설치되어 있다고 가정하자. 이와 같은 경우, 컨트롤러 컴포넌트는 특정한 지점과 외부 경계 지점에 동시에 연결되어야 하는데, 물리구조로 표현하기 어려운 문제가 발생하게 된다. 본 절에서는 이전 절에서 다루었던 열전도 해석 모델에서 컨트롤러 컴포넌트를 추가하여 온도변화를 모의하였다.

열전도를 모의하기 위한 컴포넌트는 그대로 이용하였으며, 컨트롤러 컴포넌트는 따로 구현하여, 온도 조절 규칙을 정의하였다. 개발된 컴포넌트를 이용하여 시뮬레이션 환경을 Figure 10과 같이 구축하였다. 열전도 컴포넌트는 인접하게 위치시킴으로써 서로 연결되었음을 정의하였고, 외부에 컨트롤러 컴포넌트를 위치시키고, 내부 목표 온도를 모니터링할 19번 컴포넌트와 외부조건인 3개의 컴포넌트에 연결하였다. 목표인 19번 컴포넌트의 온도가 20도를 유지하도록 3개의 외부조건 온도를 결정하는 문제로 풀이하였다.

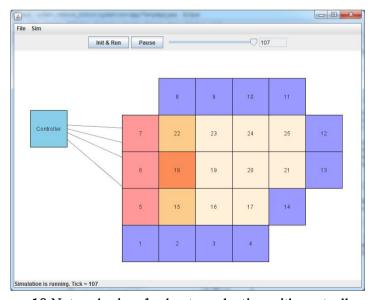


Figure 10 Network view for heat conduction with controller

풀이한 결과는 Figure 11과 같이 도시하였다. 목표 셀이 20도를 유지시키기 위하여, 컨트롤러를 이용하여 외부조건의 온도를 20도에서 100도 사이에서 온도 조절을 하도록 환경을 구성하였다. 이에 따라 그래프에서 가장 위쪽 선이 컨트롤러의 설정온도인데, 위에서 3번째 선인 목표지점의 온도와 엇갈리며 진동하는 것을 확인할 수 있다. 즉, 목표지점의 온도가 떨어지면 설정온도를 낮추고, 목표지점의 온도가 올라가면 설정온도를 올리는 방식으로 시뮬레이션이 수행되었다. 이에 따라 지점의 온도가특정한 값으로 수렴하는 현상을 확인할 수 있었다. 주어진 문제에서는 컨트롤러에서 목표온도를 77.85도로 설정하였을 때, 19번 컴포넌트의 온도가 20도로 유지됨을 확인하였다.

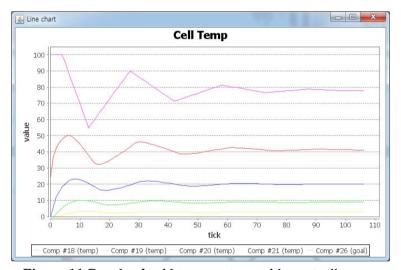


Figure 11 Result of grid temperature with controller

# 3.4 소통구조에 대한 검토

## 3.4.1 동질소통구조에서의 속성

정정 구조물인 트러스의 정역학적 부재력 해석모델은 대표적인 동질소통 시스템이다. 트러스를 구성하는 요소를 트러스와 절점으로 정의하면, 구조물을 구성하는 모든 요소는 힘의 평형상태를 유지해야 한다. 절점에 작용하는 힘과 부재에 작용하는 트러스 부재력의 합은 항상 0이다. 트러스 부재력은 트러스의 상태와 상관없이 크기만을 갖는 스칼라이다. 따라서 절점에서 평형방정식을 수립할 때 연결된 부재에서 부재력만 획득하면, 절점에서의 힘을 계산할 수 있다. 마찬가지로 부재에서는 절점의 작용하는 힘만큼을 부재에서 나눠가지면 되므로, 연결된 절점에 따라 전달율이달라지지 않는다.

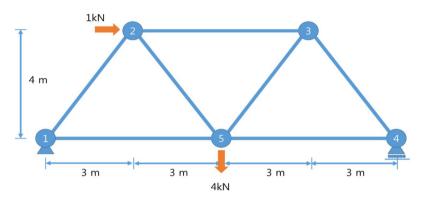


Figure 12 Problem of truss member force

트러스의 부재력을 구하는 문제가 Figure 12와 같이 주어졌을 때, 시스템 시뮬레이션을 이용하여 문제를 풀이하기 위해서는 절점 컴포넌트와 부재 컴포넌트를 개발하였다. 절점은 연결된 부재에서 부재력을 총합하여 합력을 구하고, 부재에서는 절점에 남아있는 합력을 부재력으로 치환하는 작업으로 구현하였다. 이와 같은 과정에서 각 컴포넌트는 연결된 임의의 컴포넌트에 대하여 동일한 정보교환 과정으로 구현된다.

구현한 절점 컴포넌트와 부재컴포넌트를 이용하여 트러스 부재력 해석을 위하여 Figure 13과 같이 모델을 구성하고 시뮬레이션을 실시하였다. 외부조건으로 외력이 작용하는 2, 5번 컴포넌트는 외력을 속성값으로 입력하였으며, 고정단인 1, 4번 컴포넌트는 반발력을 계산하도록 고정단임을 표시하였다. 구성된 모델을 이용하여 시뮬레이션을 실시한 결과, 연결된 컴포넌트끼리 정보를 교환하면서, 정해에 수렴하는 것을 확인하였다.

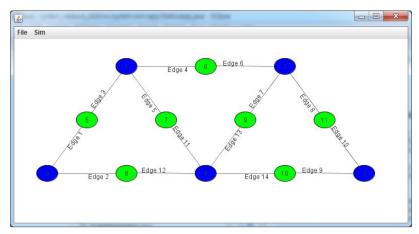


Figure 13 Simulation viewer to solve truss member force

#### 3.4.2 이질소통구조에서의 속성

이질소통구조로 연결된 속성은 컴포넌트가 연결된 컴포넌트와 정보를 교환할 때, 어떠한 경로를 선택하여 값을 취득할 것인지를 결정해야 하는 구조이다. 이와 같은 이질소통구조로 모멘트 분배법은 대표적인 문제이다. 1930년대 Cross (1930)에 의해소개된 모멘트 분배법은 컴퓨터가 발전하기 이전까지 구조해석의대표적인 방법이었다. 모멘트 분배법은 변위법의 일종으로처짐각법과 유사하며, 상대강성도와 처짐각으로부터 유도된 부재단모멘트를 미지수로 설정한다는 점에서 처짐각을 미지수로 설정하는처짐각법과 차이가 있다. 각 절점에 유도된 부재단 모멘트를 구조물전체의 평형을 이룰 때까지 반복적으로 불균형 모멘트를 분배하는방식으로, 수학적으로는 반복법을 통한 수치해석 기법이다.

여기서 시스템을 구성하는 구성요소를 절점으로 정의하였을 때, 상대강성도는 시스템을 구성하는 각 절점이 어느 절점과 연결되어 있느냐에 따라 분배율의 차이로 나타난다. 이러한 분배율은 시스템 시뮬레이션을 통하여 각 컴포넌트가 계산될 때, 같은 크기의 양 측면의 모멘트를 산정하고, 이를 초과하는 모멘트를 주변 컴포넌트에서 분배할 때 이용된다. 따라서 분배율은 컴포넌트 사이의 연결강도로 해석되며, 컴포넌트의 연결구조에 따라 서로 다른 강도를 가지게 된다. 컴포넌트의 속성은 독립적인 정보인데 반해, 연결강도는 컴포넌트의 연결 구조에 따라 서로 달라지는 정보로서, 본 연구에서는 2개 이상의 연결강도를 갖는 구조를 이질소통구조를 갖는 시스템으로 정의하였다.

이질소통구조를 검토하기 위하여, 부정적 구조물을 모멘트 분배법으로 해석하였다. 예제는 Figure 14와 같이 구성하고, 풀이 과정을 Figure 15와 같이 도시하였다(Matthews 2013). 제시한 결과와 ASAP를 이용한 시뮬레이션 결과를 비교함으로써 이질소통 구조의 재현성을 검토하였다.

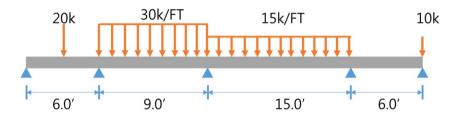


Figure 14 Problem for moment distribution method

DF	1.000	0.529	0.471	0.640	0.360	1.000	0.000
COF	0.500	0.000	0.500	0.500	0.000	0.500	0.000
FEM	-15.0	15.0	-202.5	202.5	-180.0	180.0	-60.0
BAL	15.0					-120.0	0.0
CO	0.0	7.5	0.0	0.0	-60.0	0.0	0.0
SUM	0.0	22.5	-202.5	202.5	-240.0	60.0	-60.0
DIST	0.0	95.3	84.7	24.0	13.5	0.0	0.0
CO			12.0	42.4	0.0	0.0	0.0
SUM	0.0	117.8	-105.8	268.9	-226.5	60.0	-60.0
DIST	0.0	-6.4	-5.6	-27.1	-15.2	0.0	0.0
CO			-13.6	-2.8	0.0	0.0	0.0
SUM	0.0	111.4	-125.0	238.9	-241.7	60.0	-60.0
DIST	0.0	7.2	6.4	1.8	1.0	0.0	0.0
СО			0.9	3.2	0.0	0.0	0.0
SUM	0.0	118.6	-117.7	243.9	-240.7	60.0	-60.0
DIST	0.0	-0.5	-0.4	-2.0	-1.1	0.0	0.0
CO			-1.0	-0.2	0.0	0.0	0.0
SUM	0.0	118.1	-119.2	241.7	-241.9	60.0	-60.0
DIST	0.0	0.5	0.5	0.1	0.1	0.0	0.0
СО			0.1	0.2	0.0	0.0	0.0
SUM	0.0	118.7	-118.6	242.0	-241.8	60.0	-60.0

Figure 15 Solution for moment distribution method

모멘트 분배법을 구현하기 위하여 컴포넌트는 이질소통구조로 구성된 통신방식에 따라 모델을 구현하였다. 주요한 계산과정은 주변에서 발생한 여분의 모멘트를 분배하는 과정이며, 이러한 캐리오버를 계산하는 알고리즘을 Figure 16과 같이 코드로 작성하였다. 제시한 코드 중 4번째 줄에서 "nm"은 인접한 컴포넌트를 의미하며, getDouble() 메소드를 이용하여 정보를 취득한다. 컴포넌트의 속성값은 변수의 이름으로 취득할 수 있는 반면, 연결속성은 변수 이름과 함께 현재 컴포넌트의 id를 전달하여 연결정보를 획득할 수 있도록 설계하였다.

이러한 연결정보가 모두 동일한 경우에는 따로 연결정보를 가지지 않고, 요소속성정보로 통합할 수 있다. 기존 GASS는 이러한 방식으로 동질소통구조를 구현하였다. 그러나 연결속성이 다른 경우에는 관계와 그 관계의 속성값을 저장해야 하므로, 요소속성에 비하여 관계라는 한 차원 높은 데이터 자료구조가 필요하다. 따라서 본 연구에서는 연결구조를 관리하기 위하여 컴포넌트의 속성정보에 변수명 이외에 연결대상을 2차 키로 갖는 자료구조를 설계함으로써, 이질소통구조를 표현할 수 있다.

구현된 모멘트 분배 컴포넌트를 이용하여, Figure 17과 같이 시스템을 구성하여 시뮬레이션을 구동한 결과 약 0.1%의 오차가 발생하였다. 이는 제시한 결과에서는 계산 중간에 반올림을 수행하는 반면, 시스템 시뮬레이션을 수행할 때는 중간과정에서 반올림을 하지 않고 계산하는 과정에서 생긴 차이로 판단된다.

```
01: @Override
02: public void process(List<TreeNode> neighborhoods) {
03:    for (TreeNode n : neighborhoods) {
        IASAPComponent nm = n.getModel();
        setDouble("CO", nm.getId(), nm.getDouble("BAL", getId()) * .5);
06:    }
07: }
```

Figure 16 A main code for parallel communication

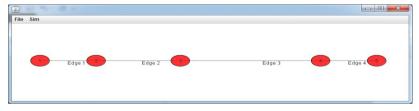


Figure 17 Simulation viewer to solve moment distribution problem

## 3.5 복제 기능에 대한 검토

# 3.5.1 복제 기능을 통한 프랙탈 구조 재현

복제 기능을 갖는 시스템은 자기 유사성이라는 특징을 갖는데, 프랙탈이라는 개념으로 설명되기도 한다. 프랙탈은 구조물의 전체형상이 시스템의 일부인 부분에서도 반복적으로 동일한 형태를 나타내는 구조를 의미하며, 단순한 규칙으로 매우 복잡한 형상을 그려낼 수 있고, 조개, 나무, 눈송이 등 자연 속에서 흔히 발견된다.

복제 기능이 가능해짐에 따라 시스템의 구조를 정적인 상태에서 동적인 상태까지 모의할 수 있게 되었다. 본 절에서는 복제 기능을 검토하기 위하여 프랙탈 구조로 표현되는 나뭇가지 성장 모델인 Lindenmayer systems를 재현하고자 한다. 식물의 생장을 모의하는 대표적인 모델로, 가지가 좌우 방향으로 번갈아생성되며, 가지의 길이는 부모 가지의 0.5배라는 간단한 공식을 이용하여 식물의 생장 형상을 제시하였다(Prusinkiewicz 1998). 이를 ASAP 컴포넌트로 개발하고, 가지의 방향을 40도로 설정하여시뮬레이션을 수행한 결과, Figure 18과 같이 성장하는 모습을 동적으로 재현하였다.

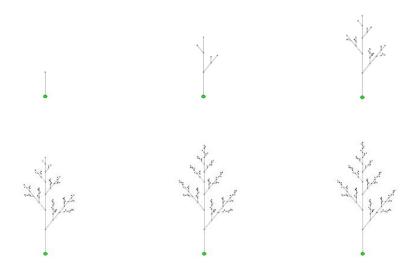


Figure 18 Simulation for lindenmayer systems

### 3.5.2 네트워크 시스템의 전처리

앞서 식물의 생장 모형을 재현하여 복제기능을 검토하였다. 이러한 시스템의 복제기능은 시스템의 외연의 확장을 모의할 수 있고, 전처리 기능으로서 시뮬레이션 구동 전에 시스템의 경계를 자동 생성하는 작업을 수행할 수 있다. 본 연구에서는 그리드 형태의 문제를 정의할 때, 다수의 컴포넌트 인스턴스를 생성하고 연결하는 과정을 최소화하기 위하여 복제기능을 이용하였다. 생성된 객체는 상하좌우로 정해진 세대만큼 복제를 반복함으로써 시스템 형상을 구성하였다.

메쉬를 생성하는 알고리즘을 구현함으로써, 지형도를 이용하여 불규칙 삼각망을 생성하는 딜라우니 삼각망 알고리즘(박미정 et al. 2006), FEM의 계산효율을 높이기 위한 적응형 메쉬 생성, 진화적최적화를 통한 구조물의 위상 최적화(이성용 et al. 2009) 등이가능한 전처리 기능으로 발전할 수 있다.

## 3.6 ASAP의 구현

### 3.6.1 시뮬레이션 환경의 구현

본 장에서 제시한 설계 및 개선사항은 반영하여 개발한 ASAP의 주요 구성을 Figure 19에 클래스 다이어그램으로 도시하였다. 시뮬레이터의 구동, 정보전달, 시각화 등 주요 업무를 관장하는 ASAPEnvironment 클래스를 중심으로 관련 클래스를 도시하였다. 시뮬레이션 프레임워크인 ASAPEnvironment 클래스는 주요 기능이 집중되어 있고, 이에 따라 가장 많은 필드와 메소드를 갖고 있다. 시뮬레이션을 실제로 구동하는 기능과 동시에 그래프를 그려주고, 시뮬레이션에 필요한 많은 클래스 사이를 중계한다.

연결된 클래스로 Type 클래스는 시스템의 연결구조를 네트워크로 나타낼 때, 방향성의 유무를 정의하는 기능을 수행한다. 컴포넌트의 속성을 이용하여 시각화하는 기능은 NetworkViewer 클래스에서 대부분의 작업을 독립적으로 수행한다. 프레임워크를 상속한 ASAPEnvironmentSqlite 클래스는 동일한 시뮬레이션 환경을 제공하면서, 모든 모의 결과를 데이터베이스로 저장하는 기능을 제공한다. 데이터베이스를 이용하면, 결과를 재현할 수 있는 장점이 있는 반면, 계산시간이 증가하는 단점이 있으므로 연구자가 상황에 따라 선택할 필요가 있다. 컴포넌트의 연결구조는 TreeNode 클래스에서, 연결정보는 Link 클래스에서 관리하며, 각 TreeNode는 시뮬레이션 모델인 IASAPComponent와 1:1로 매칭된다. 이러한 구조를 통하여 연구자가 개발한 컴포넌트는 런타임 시에 동적으로 TreeNode에 할당된다.

개발된 프레임워크는 라이브러리 형태로 제공되어, 연구자가 해당 코드를 컴파일하거나 소스 코드를 수정할 필요가 없이 클래스경로에 추가하여 이용하면 된다. 연구자가 개발한 모델 컴포넌트는 프레임워크와 별도로 관리가 가능하며, 컴포넌트 연결 및 속성을 정의한 실행용 클래스와 모델 컴포넌트 클래스 파일만 작성하여 보관하면 된다. 모델을 공유하기 위해서도 연구자가 개발한 클래스 파일만 공개하면, ASAP 환경이 갖추어진시스템에서 구동이 가능하다.

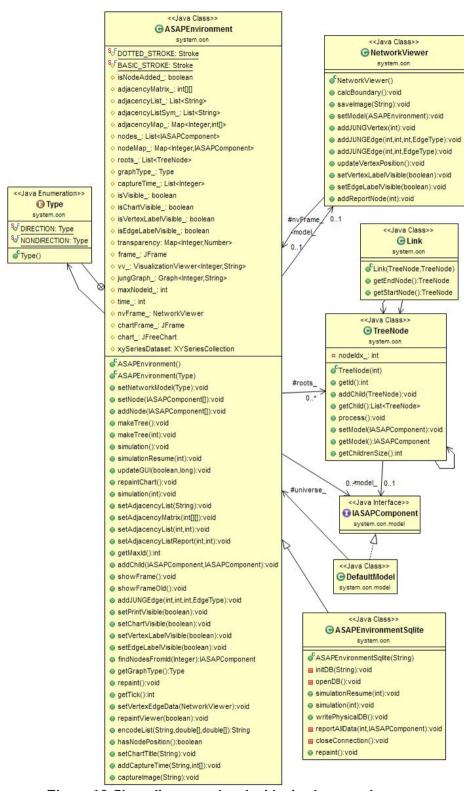


Figure 19 Class diagram related with simulator environment

#### 3.6.2 시뮬레이션 모델의 구현

시뮬레이션을 수행하기 위하여 개발한 모델 컴포넌트 및 이와 관련된 클래스를 Figure 20과 같이 클래스 다이어그램으로 도시하였다. 컴포넌트 클래스를 중심으로 관계도를 살펴보면, 그림의 가운데 있는 IASAPComponent 인터페이스가 가장 중요하다. 이 클래스를 구현하면, 기본적으로 ASAP 환경에서 모델을 구동할 수 있다.

인터페이스에는 정보를 교환하기 위한 기능뿐만 아니라, 화면에 컴포넌트 노드를 그리기 위한 속성이나, 종료조건, 시뮬레이션 환경에 대한 레퍼런스 등 다양한 메소드가 존재하고, 대부분의 메소드는 모델마다 중복된다. 따라서 인터페이스의 공통된 기능을 구현한 추상클래스인 DefaultModel를 제공하였다. 모델 컴포넌트를 개발하기 위해서는 DefaultModel를 상속하여 모델에 필요한 지배방정식을 java 코드로 작성하여 구현하면 된다.

클래스 다이어그램에는 본 연구에서 개발한 다양한 모델 컴포넌트도 확인할 수 있다. 각 모델 컴포넌트 클래스는 DefaultModel를 상속하며, 생성자, initialize(), process(), isComplete() 메소드를 공통적으로 오버라이드하여 구현하고 있다. 모델에 따라 추가적인 메소드가 존재하나, 대체적으로 메소드 구성이 비슷한 것을 확인할 수 있다.

모델 컴포넌트와 연결된 클래스를 살펴보면, DefaultModel 클래스 외에는 연결된 클래스가 존재하지 않으며, 시뮬레이션 환경인 ASAPEnvironment 클래스 및 연결 정보를 관리하는 TreeNode와는 직접적으로 연결되지 않고, DefaultModel 클래스와 IASAPComponent 인터페이스를 경유하여 연결되어 있음을 확인할수 있다. 이러한 구조는 연구자가 만든 모델 컴포넌트 클래스와 시뮬레이션 환경 사이에 느슨하게 결합함으로써 개방형 구조를 제공하고자 하였다. 즉, 연구자가 개발한 모델을 추가하거나 삭제하더라도 ASAP를 구동하는 과정에서 영향을 미치지 않으므로 오류가 발생하지 않고 정상 작동이 가능하다.

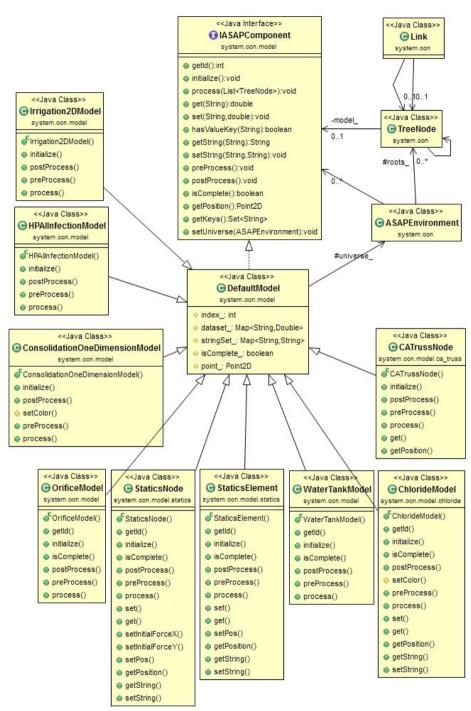


Figure 20 Class diagram related to implement system models

# 제 4 장 지역시스템 문제의 적용

# 4.1 서론

시스템 시뮬레이션은 시스템 전체를 설명하는 연립방정식을 수립하지 않고, 개별 요소의 지배방정식을 기술함으로써 전체 거동을 모의하는 데 목적이 있다. 개발한 ASAP은 물리구조를 갖는 시스템뿐만 아니라 위상구조를 갖는 시스템을 모의할 수 있으며, 동질소통구조로 연결된 컴포넌트 혹은 이질소통구조로 연결된 구조에 대해서도 모의할 수 있다고 기술하였다. 또한 복제 기능을 통하여 성장하는 시스템 역시 모의가 가능하다. 이러한 개선 기능을 중심으로 요소 컴포넌트를 개발하고, 이를 연결하여 시스템을 모의하고자 한다.

개발한 시스템의 효용을 평가하기 위하여 다양한 적용 예의 선정이 필요하다고 판단한 바, ASAP의 주요한 대상분야인 지역시스템과 관련하여 세부 전공 별로 문제를 선정하고자 하였다. 수리 분야, 구조 분야, 토질 분야, 관개 분야, 재료 분야, 방재 분야로 세부 분야를 설정하고, 각 분야별로 예제를 준비하였으며, 선정된 적용 예제는 Table 3과 같이 정리하였다. 전체 9가지 모델을 적용하였으며, 이중 다수의 예제가 동질소통구조로 연결된 물리적으로 결합한 문제로 분류된다. 이는 기존 GASS로 선정한 예제 중 상당한 시스템을 모의할 수 있음을 의미하며, ASAP를 통하여 앞서 지적한 연결, 소통, 복제 측면에서 기능을 향상함으로써 적용 분야를 확장하고, 유틸리티 제공을 통하여 사용편의성을 개선한 것으로 평가할 수 있다.

Table 3 Classification of example problems

	Topological link	Physical link
Homogeneous	Water Tank Consolidation	
	Three-reservoir	Soil moisture
		Chloride
Heterogeneous	Infection	Truss
Cloning		Growth of plant
		Equivalent truss

본 장에서는 분야별로 선정된 문제를 적용하는 데 있어서 학계에 발표되어 검증된 연구 결과를 재현하는 방식으로 다수의 적용 연구를 진행하였다. 이러한 접근을 통하여 기존 연구결과를 본시스템 시뮬레이터를 통해 구현할 수 있음을 검증하고, 간편한 구현 방법과 기존 분석결과와 일치하는 시뮬레이션 결과를 제시함으로써 시스템 시뮬레이터의 효용을 설명하고자 하였다. 앞으로 여러 가지모델이 컴포넌트로 구현되어 축적되면, 연구자는 모델 개발 과정을 생략할 수 있으므로, 컴포넌트를 생성하고, 연결하고, 속성을 부여하는 시뮬레이션 환경을 구축하는 절차만으로 시스템 시뮬레이션이 가능하므로 본 장에서 제시한 적용 예에 비하여 간소하게 ASAP를 이용할 수 있을 것으로 기대한다.

## 4.2 모델 구현과 구동

모델을 구현하기 위해서는 앞서 설명한 DefaultModel를 상속하여 컴포넌트 클래스를 작성해야 한다. 본 장에서는 각 모델의 메커니즘을 기술하는 주요한 코드만을 제시하였으나, 해당 코드만으로도 컴포넌트를 재현하는 데는 어려움이 없을 것으로 판단하였다.

컴포넌트를 작성하였거나, 작성된 컴포넌트를 가지고 있다면, 모델 컴포넌트를 이용하여 시뮬레이션을 구동하기 위한 간단한 스크립트 프로그램을 작성해야 한다. 시뮬레이션 구동을 위한 프로그램은 다음과 같은 순서로 작성된다.

- 1) 시뮬레이션 환경인 ASAPEnvironment 인스턴스를 생성
- 2) 컴포넌트 IASAPComponent 인스턴스를 생성
- 3) 컴포넌트 인스턴스의 속성 지정
- 4) 컴포넌트 인스턴스를 시뮬레이션 환경에 등록
- 5) 컴포넌트 인스턴스의 연결 정보를 시뮬레이션 환경에 등록
- 6) 시뮬레이션 환경을 초기화하고, 시뮬레이션 실행

위 순서에 따라 작성된 "연결된 저수조 2개의 수위변화 모의"에 관한 코드를 Figure 21과 같이 도시하였다. 먼저, 시뮬레이션 프레임워크인 ASAPEnvironment 인스턴스 생성과정을 4-6줄에 작성하였다. 데이터베이스 이용 여부에 따라 4번째 줄 혹은 5번째 줄 코드를 선택하고, 데이터베이스를 이용하는 경우에는 데이터베이스의 경로를 매개변수로 넘겨주어야 한다는 점에 주의해야 한다.

다음으로 컴포넌트 인스턴스를 생성하는 과정이 7-12번째 줄까지의 코드이다. 여기서는 2개의 OrificeModel 인스턴스를 만들고, "height" 변수를 그래프로 관찰하도록 지시하였다. 생성된 컴포넌트 인스턴스에 속성을 지정하는 과정이 13-14번째 줄이다. 이로써 컴포넌트 준비가 완료되었고, 이를 ASAPEnvironment에 전달하여 프레임워크에서 구성요소를 인지하도록 설정하였다. 시스템을 구성하는 요소들 간의 연결 구조는 17번째 줄과 같이 작성하여, 시뮬레이션 환경에 전달한다. 남은 코드는 주어진 정보를 바탕으로 시뮬레이션 환경을 초기화하고, 값을 추적하기 위한 그래프와 문제를 시각화하여 보여주는 NetworkViewer가 화면 상에 나타나도록 명령을 수행한다. 마지막으로 각 컴포넌트 별로연결구조를 확인하고, 시뮬레이션이 수행된다. 이러한 코드는 간단하게 작성할 수 있으며, 향후 GUI (Graphical User Interface)로 대체될 수 있다.

```
01:
     public class OrificeApp {
02:
         public static void main(String[] args) {
     //
             if no database, uncomment
03:
             ASAPEnvironment app = new ASAPEnvironment();
     //
04:
             ASAPEnvironment app = new
05:
                  ASAPEnvironmentSqlite("db/orifice.db");
06:
07:
             int nodeCount = 2;
08:
             IASAPComponent[] nodes = new IASAPComponent[nodeCount];
09:
             for (int i = 0; i < nodeCount; i++) {</pre>
10:
                nodes[i] = new OrificeModel(i+1);
11:
                nodes[i].setString("plot_item", "height");
12:
             nodes[0].set("height", 50);
13:
             nodes[1].set("height", 200);
14:
15:
             app.setNode(nodes);
16:
             app.setAdjacencyList(nodes[0].getId(), nodes[1].getId());
17:
             app.setChartTitle("Heights");
18:
             app.showFrame();
19:
             app.makeTree();
20:
             app.simulation();
21:
         }
22:
     }
```

Figure 21 A main code for setting simulation environment

## 4.3 동질소통을 하는 위상구조

### 4.3.1 문제 선정

시스템 시뮬레이션은 시스템을 구성하는 요소가 서로 영향을 주고 받는 현상을 모의하기 위한 방법으로, 수식으로 설명하기 어려운 문제를 해결하기 위하여 사용된다. 본 절에서는 단순하지만. 수식으로 풀기에 복잡한 문제로 단일 변수를 갖는 미분 문제를 선정하였다. 연결된 저수조의 수위 문제는 수위를 단일 변수로 정의하여, 시간에 따라 변화하는 수위를 해석하는 문제이다. 저수조를 컴포넌트로 정의하여, 수위를 속성값으로 가지며. 저수조를 연결하는 관의 굵기는 동일하다고 가정하였다. 저수조의 수위는 오로지 연결된 수조 간의 위치에너지만 고려하며, 연결관에서의 손실수두는 무시하였다. 이와 같은 가정에 의하여 저수조의 위치는 수두변화에 고려되지 않으므로, 위상구조 문제에 해당하며, 수조관을 연결하는 관의 굵기가 동일하므로 동질소통으로 구성된 문제로 분류된다. 선정된 문제는 미분 방정식의 해석해와 시스템 시뮬레이션을 통한 분석 결과를 비교함으로써 다수의 요소가 결합된 시스템의 해석방법으로 시스템 시뮬레이션의 적용 가능성을 살펴보고자 하였다.

## 4.3.2 연결된 저수조의 수위해석

물이 차 있는 원통을 파이프로 연결할 경우, 서로 연결된 물통사이의 위치에너지 차이에 의하여 물이 흘러 종국에는 연결된 모든물통의 수위가 같아진다. 이와 같은 현상을 분석하기 위하여, 미분방정식의 풀이를 통한 해석해(Kreyszig 2010)를 계산하고, 이를 ASAP를 이용한 시뮬레이션 결과와 비교하였다. 원통형의 저수조는 Figure 22와 같이 도시할 수 있으며, 저수조 가장 아래부분에 유출구가 위치한다. 유출구가 개방되어 물이 흘러나옴에따라 변화하는 저수조의 수위 함수 h(t)를 구하는 문제로서, 유출구의 단면적을 A. 원통 수조의 밑변 면적을 B로 정의하였다.

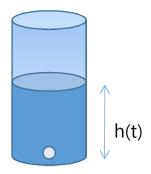


Figure 22 Problem for water height function of one tank

위와 같은 조건에서 저수조의 유출구로 분출되는 물의 유출속 도는 위치에너지가 운동에너지로 전환된 상태이므로, 식 (2)와 같이 정리된다.

$$v(t) = C\sqrt{2gh(t)} \tag{2}$$

여기서 C는 유출구의 크기 및 형태에 따른 형상계수인데, 본 연구에서는 이상적인 상태로 가정하여 C=1로 정의하였다. 유출속도에따라 유출되는 물의 부피 V는 식 (3)과 같이 유출구 단면적과 유출속도의 곱인 시간당 유출량에 단위 시간을 곱하여 정리된다.

$$\Delta V = Av(t)\Delta t \tag{3}$$

저수조 안의 부피변화는 식 (4)와 같이 저류조 단면적에 줄어든 수 위변화량의 곱으로 정리된다.

$$\Delta V^* = -B\Delta h \tag{4}$$

여기서 질량보존법칙에 따라, 유출된 물의 양과 저수조 안에서 줄어든 물의 양은 동일해야 하므로, 식 (3), (4)를 등치하고, 식 (2)를 이용하여, 유출 속도항을 수위함수로 표현되는 우항으로 치환하여다음 식 (5)와 같이 정리할 수 있다.

$$Av\Delta t = -B\Delta h$$

$$v = -\frac{B}{A}\frac{\Delta h}{\Delta t}$$

$$\sqrt{2gh} = -\frac{B}{A}\frac{\Delta h}{\Delta t}$$

$$\frac{1}{\sqrt{h}}\Delta h = -\frac{A}{B}\sqrt{2g}\Delta t$$
(5)

양변을 각각 h 와 t 로 적분하고, 초기조건  $h(0) = H_0$  라고 가정하였을 때, 식 (6)과 같이 수위 함수 h(t) 의 해석해를 도출하였다.

$$h(t) = \frac{1}{2} \left( -\frac{A}{B} \sqrt{g}t + \sqrt{2H_0} \right)^2 \tag{6}$$

#### 1) 연결된 저수조의 수위 함수

문제를 확장하여, 두 개의 저수조가 연결 연결되어 있는 경우, Figure 23과 같이 도시할 수 있다. 두 개의 저수조가 연결된 경우에는 위치에너지가 두 저수조의 수위 차에 의해 결정되므로, 앞서수위 함수에서 두 저수조의 수위관계가 추가된다. 연결 파이프의 단면적은 A, 두 저수조의 밑면 면적은 B로 동일하다고 가정하였으며, 수치해석 결과와 비교하기 위하여, 저수조의 초기 수위는 각각  $h_1(0) = 50$ ,  $h_2(0) = 200$ 으로 가정하였다.

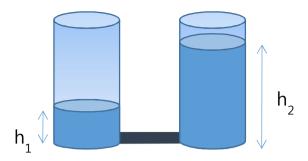


Figure 23 Problem for water height function of two tanks

수위가 높은 2번 저수조를 기준으로 유출 속도는 다음과 같이 두 저수조의 수두차를 이용하여 정리할 수 있다.

$$v = \sqrt{2g\left(h_2(t) - h_1(t)\right)} \tag{7}$$

전체 물의 양은 보전되고, 두 수조는 같은 형상으로 구성되어 있으므로, 수두를 이용하여 보전방정식을 식 (8)과 같이 정리할 수 있다.

$$h_1(t) + h_2(t) = 250 (8)$$

여기에 시간에 따른 수두 변화는 유출속도와 비례하므로 다음과 같이 저수조의 수위와 연결 파이프의 유속으로 정리된다.

$$\Delta h_1(t) = \frac{A}{B} v \Delta t \tag{9}$$

식 (8)-(9) 의 관계식을 정리하면, 식 (10)과 같이 전개된다.

$$\frac{1}{\sqrt{250 - 2h_1(t)}} \Delta h_1(t) = \frac{A}{B} \sqrt{2g} \Delta t \tag{10}$$

양변을 적분하여 식 (11)를 구하였다.

$$\sqrt{250 - 2h_1(t)} = \frac{A}{B}\sqrt{2g} \cdot t + C \tag{11}$$

초기조건을 입력하여, 두 저수조의 수위 함수  $h_1(t)$ 와  $h_2(t)$ 에 대하여 식 (12), (13)과 같이 구하였다.

,

$$h_{1}(t) = \frac{250 - \left(-\frac{A}{B}\sqrt{2g} \cdot t + \sqrt{150}\right)^{2}}{2}$$
(12)

$$h_2(t) = 250 - h_1(t) \tag{13}$$

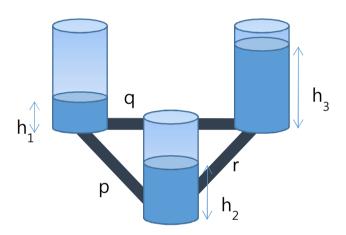


Figure 24 Problem for water height function of three tanks

문제를 Figure 24에 도시한 바와 같이 2개의 저수조에서 3개의 저수조로 확장하였다. 저수조의 개수가 증가함에 따라 관계식은 1개이던 평형방정식이 3개로 늘어나고, 특히 저수조의 특성인 밑면적이나 연결 파이프의 두께가 다를 경우, 해석해는 더욱 복잡해진다.

초기조건  $h_1(0)=50$ ,  $h_2(0)=150$ ,  $h_3(0)=100$ 으로 가정하고, 각 각의 단면적을  $A_p$ ,  $A_q$ ,  $A_r$ 로 정의하여 지배방정식을 정리하면 다음 식 (14)와 같다. 주어진 식은 비선형 미분 연립 방정식으로 해석적으로 풀이하기가 매우 어렵다.

$$h_{1}(t) + h_{2}(t) + h_{3}(t) = 200$$

$$\frac{dh_{1}(t)}{dt} = \frac{A_{p}}{B} \sqrt{2g(h_{3}(t) - h_{1}(t))} + \frac{A_{q}}{B} \sqrt{2g(h_{2}(t) - h_{1}(t))}$$

$$\frac{dh_{2}(t)}{dt} = -\frac{A_{q}}{B} \sqrt{2g(h_{2}(t) - h_{1}(t))} - \frac{A_{r}}{B} \sqrt{2g(h_{2}(t) - h_{3}(t))}$$

$$\frac{dh_{3}(t)}{dt} = -\frac{A_{p}}{B} \sqrt{2g(h_{3}(t) - h_{1}(t))} + \frac{A_{r}}{B} \sqrt{2g(h_{2}(t) - h_{1}(t))}$$
(14)

#### 2) 저수조 컴포넌트의 구현

앞서 미분방정식의 풀이 과정을 통하여, 저수조의 수위함수를 계산하였다. 단순한 현상임에도 불구하고, 저수조의 개수가 늘어날수록점차 해석하기 어려운 단계에 이름을 확인하였다. 시스템 시뮬레이션은 이와 같이 복잡한 시스템 현상에 대하여, 앞서 미분방정식의풀이와 같이 전체에 대하여 모델링하는 방식과 달리, 시스템을 구성하는 요소 단위로 분석하여 모델링하고, 개개의 요소를 통합함으로써 시스템의 거동을 모의한다.

저수조가 여러 개가 연결되는 시스템 문제는 저수조에서 수위 량 변화 거동 규칙, 즉 위치에너지가 운동에너지로 변환되는 지배방 정식은 동일하지만, 여러 개가 결합하는 과정에서 수위 함수가 서로 연관성이 가지게 되면서 복잡도가 증가하게 되었다. 시스템 전체를 이해해야만 문제를 해결할 수 있는 공학적 해석방법의 특징이 잘나타난 현상이다. 반면 ASAP는 단위요소의 거동규칙을 정의하여, 단위요소를 연결시키는 작업만으로 전체의 현상을 모의할 수 있고, 이를 통해 문제를 손쉽게 해결할 수 있다.

저수조의 지배방정식은 연결된 수조와 상호간의 수위차에 의한 위치에너지를 운동에너지로 변환하는 과정이므로, 식 (2)를 기반으로 모델링하였다. 미분방정식은 시간 흐름에 따라 모의하는 ASAP 특성에 따라 차분식 형태로 식 (15)와 같이 정리하여 적용하였다.

$$h_1(t) = h_1(t-1) + \sum_{k=2}^{n} \frac{A}{B} \sqrt{2g(h_k(t-1) - h_1(t-1))} \times \Delta t$$
 (15)

ASAP 환경에서 컴포넌트 구현과정 중 주요 부분을 Figure 25

와 같이 도시하였다. 코드에서 process() 메소드에서 차분식을 이용하여, 단위 시간동안 수위변화량을 계산하였다. 연결된 요소로부터 수위값을 가져와서 비교하는데, 코드에서 알 수 있듯이 연구자는 코드상에서는 연결된 노드가 무엇인지 알 수 없고, 단지 "neighborhoods"라는 노드 리스트를 매개변수로 전달받아서 이용한다. 이 리스트는 사용자가 문제를 정의하고, 시스템을 구성하는 컴포넌트를 연결해준 정보를 바탕으로 런타임 시에 비로소 제공된다. 코드 하단에 postProcess() 메소드에서 이전 수위와 수위변화량을 더하여 값을 갱신하는 것을 확인할 수 있다. 이와 같이 단계를 분리한 이유는 process() 메소드는 각 컴포넌트가 동시에 수행되므로, 이 과정에서 서로 참조하는 값이 변경되면, 컴포넌트에 따라서서로 다른 시점의 값을 참조하게 되므로 동일한 시점인 t-1의 값을 이용하여 계산되도록 하기 위함이다.

```
01:
     @Override
02:
     public void process(List<TreeNode> neighborhoods) {
03:
         double h1 = getDouble("height");
04:
         double transferTotal = 0;
05:
            for (TreeNode n : neighborhoods) {
06:
                IASAPComponent nm = n.getModel();
07:
                double h2 = nm.getDouble("height");
08:
                double trans = Math
09:
                    .sart(2 * 9.81 * Math.abs(h2- h1)) * .01;
10:
                if (h1 > h2) trans = -trans;
11:
                transferTotal += trans;
            }
12:
13:
14:
         setDouble("trans", transferTotal);
15:
     }
16:
17:
     @Override
18:
     public void postProcess() {
19:
         setDouble("height", getDouble("height") + getDouble("trans"));
20:
```

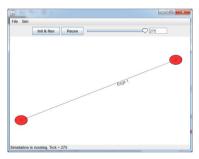
Figure 25 Program code for water height problem with linked tanks

행렬식을 이용하여 반복해석을 수행하는 모델은 ASAP 환경으로 비교적 쉽게 변환할 수 있는데, 이는 ASAP의 계산과정이 행렬을 자코비 반복법으로 행렬을 계산하는 과정과 유사하기 때문이다. 행렬에서는 차례대로 방정식을 계산하는 반면, ASAP 환경에서는 순서를 보장하지 않는다는 점만 다르다. 특히, 개발한 컴포넌트에서는 process()와 postProcess()를 분리하여 모델링을 수행하였는데,

이를 분리하지 않고 구현하면 가우스 세이델법과 유사하게 형태로 문제를 풀이하게 된다. 즉, 참조하는 값이 이미 계산되었을 경우, 이 전 값을 사용하지 않고 갱신된 값을 이용하도록 구현하면 보다 효 과적으로 수렴하도록 풀이할 수 있다.

#### 3) 연결된 저수조의 수위 해석

구현한 컴포넌트를 이용하여, ASAP 환경에서 연결 정보 및 초기속성 정보를 입력하면 시스템 시뮬레이션이 가능하다. 초기 조건으로는 각 저수조의 초기 수위가 필요하며, 각 저수조는 방향성이 없이 완전 연결된 형태로 네트워크를 구성하였다. 우선적으로 2개의컴포넌트를 연결하여, 저수조가 2개일 경우를 모의하였다. 모의 결과는 해석해와 비교함으로써 결과를 검증하였다.



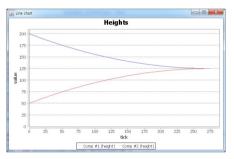


Figure 26 Screenshots of ASAP for two tanks problem

모의 결과는 Figure 26과 같이 시스템의 네트워크 구성도와 시뮬레이션 결과 그래프를 실시간으로 표시된다. 저수조 2개는 각각 2개의 붉은 원으로 표시되며, 두 원이 선으로 이어져 있는 것을 확인할 수 있다. 선이 연결된 컴포넌트 간에만 정보교환이 가능하며, 이후 다른 예제를 통하여 다루도록 하겠다. 시뮬레이션 결과 그래프는 컴포넌트에서 지정한 속성만 관찰할 수 있으며, 선택된 속성은 시뮬레이션 시간에 따라 변화되는 값을 동적으로 확인할 수 있다.

모의 결과를 검증하기 위하여, 미분방정식을 풀이한 해석해와 시뮬레이션 결과를 Figure 27에 도시하였다. 그래프에서 확인할 수 있듯이 모의결과가 해석해와 동일하다고 판단할 수 있다. 그러나 차 분식을 기반으로 모의하기 때문에 단위시간을 크게 설정하면 오차 가 발생할 수 있으므로 주의가 필요하다.

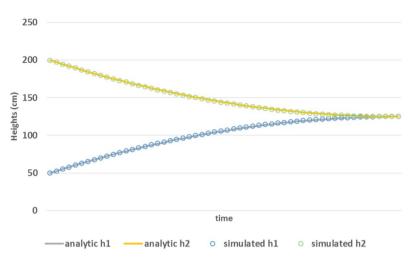


Figure 27 Comparison of simulated result to analytic result

차분식에 사용되는 단위시간의 크기에 따라 모의 결과의 오차를 비교하기 위하여, 두 저수조의 수위가 동일해지는 시점을 기준으로 Figure 28에 비교 정리하였다. 결과에서 점선은 해석해이고 점으로 표시된 자료는 시뮬레이션을 통해 모의된 시간을 의미한다. 해석해의 결과는 2.766 시간에 동일한 수위를 갖게 되는데, 1시간과 0.5시간으로 모의했을 경우는 동일 수위가 되는 시점이 3시간인데반해, 0.1시간 단위로 모의했을 경우, 2.7시간, 0.01시간 단위로 모의할 경우 2.75시간으로 산정되었다. 이는 단위 시간이 작아질수록정해에 수렴해가는 과정으로 판단할 수 있다. 해석의 정확도를 위해서 단위시간을 작게 설정해야 그러나 단위 시간의 크기는 전체 계산시간과 반비례하므로, 연구자가 적절한 단위 시간을 선정할 필요가 있다.

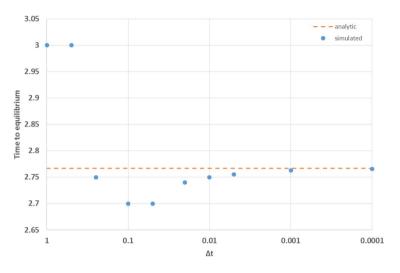


Figure 28 Simulation model results related to delta time

서브 시스템을 결합하여 시스템을 모의하는 시스템 시뮬레이션은 모델을 확장할 때 효용성이 드러난다. 앞서 검증한 모델을 바탕으로 저수조를 3개로 늘려서, 0.01시간 단위로 시뮬레이션을 수행한결과를 Figure 29와 같이 도시하였다. 초기 조건은  $h_1(0)=50$ ,  $h_2(0)=150$ ,  $h_3(0)=100$ 로 설정하여, 저수조의 수위가 같아지기까지는 1.67 시간이 걸리는 것으로 모의되었다. 저수조별 수위를 살펴보면, 1.45 시간이 경과한 시점에 첫 번째 저수조와 세 번째 저수조의 수위가 115.91 cm로 같아진 후, 전체 저수조의 수위가 116.67 cm로 동일할 때까지 수위 변화가 동기화된 행태를 확인하였다.

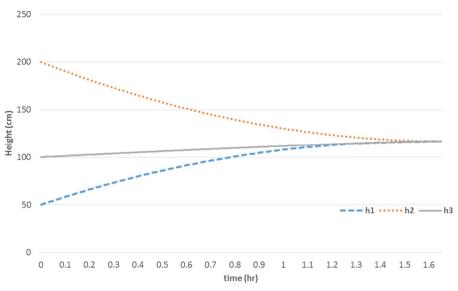


Figure 29 Simulated water height for three tanks problem

서로 연결된 저수조의 수위 변화를 모의한 결과를 통하여, 미분 연립방정식을 풀이하는 방법에 비하여 연구자가 쉽게 현상을 분석 할 수 있음을 확인하였다. 본 연구에서는 3개의 저수조만 연결하였 지만, 시뮬레이션 모델 구성을 고려하면 n개의 저수조로 확장 가능 하다고 판단된다. 또한 위상구조로 저수조의 연결을 정의하였기 때 문에 2차원 그리드 형태가 아닌 복잡한 형태의 네트워크로 구성된 연결을 갖더라도 문제를 정의할 수 있다.

앞서 식 (14)와 같이 연결관의 면적이 달라지는 경우에는 동일 시간 내에 전달량이 달라지게 된다. 이 경우에는 대상 저수조에서 어느 저수조와 연결되는 지에 따라 행위가 달라지는 형태, 다시 말 해 연결되어 있는 그룹 중 어느 한 경로를 선택해야 하는 이질소통 구조의 문제로 치환된다. 이질소통구조를 갖는 문제는 동질소통구조 를 갖는 문제와 동일하게 컴포넌트를 구현하되 정보를 가져오는 부 분에서 이질소통구조로 기술하고, 모델 적용 시에 이질소통에 따른 개별 속성을 입력하여 정상적으로 시스템 시뮬레이션이 가능하다.

### 4.3.3 삼형제 저수지의 수위 모의

앞서 풀이한 연결된 저수조의 수위 모형을 활용하여, 삼형제 저수지의 수위를 모의하였다. 삼형제 저수지는 서로 인접한 3개의 골짜기를 막아서 축조되었으며, 서로 파이프로 연결되어 관개용수와 관개지역을 공유하는 특징이 있어 모델링 측면에서 연결된 저수조의 수위 모형과 매우 유사하다. 본 절에서는 저수조 모형에 강우에 따른유입량, 관개량, 증발량, 침투량 등을 고려한 저수지 물수지 모형으로 확장하여 저수지에서 실측된 수위 자료와 비교하고자 하였다. 분석에 필요한 제원 및 모형을 검토하고, 점진적으로 시스템 모델을확장하여 최종적으로 삼형제 저수지의 수위를 모의하였다.

삼형제 저수지는 충청북도 음성군에 위치한 용계, 무극, 금석 3 개의 저수지를 의미한다. 다음 (2014)에서 제공하는 지도 서비스를 이용하여 Figure 30에 인근 지형도를 도시하였으며, 위쪽부터 금석 저수지, 무극저수지, 용계저수지 순이다. 관개현황을 살펴보면, 용계 저수지를 통해서 774.6 ha의 수혜지역에 농업용수를 공급하며, 금석저수지에서는 102.5 ha의 지역에 용수를 공급하고 있다. 무극저수지는 인접한 용계저수지와 금석저수지에 용수를 공급하며, 직접적으로 관개를 실시하지는 않는 것으로 조사되었다.



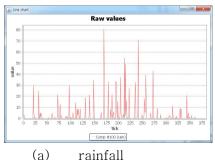
Figure 30 Map of three reservoirs

먼저, 강우에 따른 유역 유출량을 산출하여, 저수지의 유입량을 모의하고자 하였다. 대상 지역의 관할 측후소는 티센망도에 따라 충 주 기상청이 지배 관측소로 조사되었다. 기상청 (2014)에서 제공하 는 기상자료를 수집하여 기상자료 컴포넌트를 구현하였으며, 유역 유출량을 모의하기 위해서 3단 탱크 모형을 이용하여 유역 컴포넌 트를 개발하였다. 탱크 모형의 매개변수는 김현영 and 박승우 (1988)이 소규모 저수지에서 사용할 수 있도록 제안한 회귀식을 이용하였으며, 회귀식에 사용한 유역 제원은 Table 4와 같다.

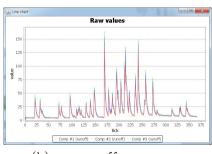
Table 4 Watershed characteristics of three reservoirs

	Geumseok	Moogeuk	Yonggye
Paddy (%)	5.9	3.1	7.5
Forrest (%)	76.5	83.0	74.4
Upland (%)	7.5	5.5	11.4
Area (km²)	5.3	6.3	12.1

기상자료 컴포넌트에서 강수량 자료와 유역 컴포넌트에서 강수 량 자료를 이용하여 계산된 유출량을 모의한 결과 Figure 31과 같 이 결과를 확인할 수 있다. 유출량은 3개의 유역 특성을 고려하여 매개변수를 결정하였으므로, 유출패턴이 동일하지 않으나, Figure 31 (b)에서는 동일한 기상자료를 이용하여 장기간 유출량을 모의하 였기 때문에 3개의 유역에서 거의 동일한 패턴의 유출이 발생하는 것으로 관찰된다. 개발된 유역 컴포넌트는 기상자료 컴포넌트에 연 결하면, 해당 측후소의 강우량에 따라 유출량을 계산하므로, 단위도 법과 같은 원리로 유사 연구에서 재이용 가능성이 높을 것으로 기 대한다.







(b) runoff

Figure 31 Simulated results of weather and watershed components

저수지 컴포넌트는 앞서 저수조 컴포넌트를 개량하여, 유입량, 증발량, 침투량을 고려하여 저류량을 결정하도록 개선하였다. 그리고 저수지는 수위에 따라 내용적이 달라지므로, 수위-내용적 테이블을 구축하고, 내삽을 통하여 수위-내용적을 계산하였다. 계산에 이용한 수위내용적은 Table 5와 같다.

Table 5 Level-storage (m-10<sup>3</sup>m<sup>3</sup>) lookup table for three reservoirs

Level (m)	Geumseok	Moogeuk	Yonggye
113	0.0		
114	0.8		0.0
115	4.5		1.5
116	12.3		8.1
117	26	0.0	23.8
118	48.7	0.7	50
119	81.9	4.2	87.2
120	130.1	11.7	141.1
121	197.9	24.8	217.1
122	287.8	51.4	312.8
123	409.5	97.2	427.3
124	561.4	161.7	573.9
125	744.7	243.1	757.1
126	952.9	338.5	978.1
127	1185.5	454.3	1229.2
128	1452.5	598.8	1509.7
129	1757.3	773.3	1831.7
130	2113.3	971.5	2199.2
131	2516.1	1192.8	2614.1
132	2976	1454.1	3094.6
133	3512.6	1840.9	3667.0

134	4098.1	2294.1	4297.9
135	4732.0		

관개지역에서는 필요한 관개량을 계산하기 위하여, FAO Penman 수정식(Doorenbos and Pruitt 1977)을 이용하여 잠재증발산량을 산정하고, 유승환 et al. (2006)가 제시한 논벼 작물계수를 곱하여 증발산량을 계산하였다. 산정된 증발산량에 관개효율을곱하여 필요수량을 소비하는 관개지역 컴포넌트를 구현하였다. 이상에서 살펴본 컴포넌트와 저수지의 제원을 바탕으로 Figure 32와 같이 삼형제 저수지의 수위 모의 시스템을 구성하였다. 기상자료 컴포넌트는 유역 컴포넌트와 관개지역 컴포넌트 모두 관계되므로, 필요한 관계를 정의하여 시스템을 구성하였다.

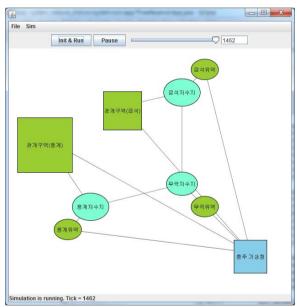


Figure 32 Water level simulation for linked three reservoirs

구성한 시뮬레이션 모델을 이용하여, 2010년부터 2013년까지 4개년도 저수위를 모의한 결과, Figure 33과 같이 수위 변화를 모의하였다. 수위 변화가 가장 큰 선이 대부분의 관개용수를 공급하는 용계저수지의 수위이며, 관개시기를 제외하고는 삼형제 저수지의 수위가 거의 패턴으로 거동하고 있음을 확인할 수 있다. 관개시기에는 관개수요가 많은 용계저수지의 수위가 급격히 하락하고, 이후 수일

동안 무극저수지로부터 용수를 공급받아 수위를 회복하는 것을 확 인할 수 있다.

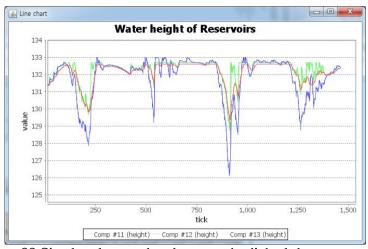
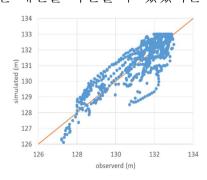
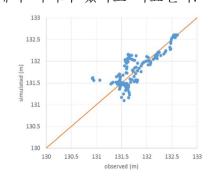


Figure 33 Simulated water level among the linked three reservoirs

모의된 수위를 검증하기 위하여 실측 수위자료와 비교하였다. 한국농어촌공사는 자사 관할 저수지를 대상으로 실시간 수위계를 설치하여 운영하고 있으며(남원호 and 최진용 2013), 삼형제 저수지 중에서 용계저수지와 금석저수지의 수위를 모니터링하고 있다. 용계저수지는 2010년 1월 1일부터 자료를 구득할 수 있었으며, 금석저수지는 2012년 11월 9일부터 2013년 10월 21일까지 자료를 구득할 수 있었다. 구득한 실측자료와 ASAP를 통해 모의한 결과를 비교한 결과 Figure 34와 같이 1:1 그래프로 도시하였다. 결정계수는 각각 0.888, 0.857로 분석되었고, RMSE는 각각 0.92 m, 0.19 m로 나타났다. 실측 값을 매우 정확하게 추정한다고는 판단할 수 없으나, 다양한 인자를 고려하여 점진적으로 모델을 확장하여, 일정한 패턴을 확인할 수 있었다는 점에서 의의가 있다고 사료된다.





#### Figure 34 Scatter plot between observed and simulated level

## 4.4 동질소통을 하는 물리구조

### 4.4.1 문제 선정

동질소통을 하는 물리구조로 표현되는 문제는 연속된 요소사이에서 등방성을 갖는 특성으로, 일반적으로 유한차분법(Finite Difference Method, FDM)으로 풀이하는 문제에 해당한다. 유한차분법은 미분방정식의 미분항을 유한차분으로 치환하여풀이하는 수치해석방법으로, 테일러 급수를 이용하여 미분항을 변환한다. 본 절에서는 유한차분법으로 설명되는 다양한 예제를 선정하였다. 토질 분야에서는 연직배수를 고려한 1차원 토질압밀을, 관개분야에서는 점적관개에 의한 토양수분의 분배를, 재료분야에서는 해양구조물인 배수갑문의 염해를 대상 문제로 선정하여모의하였다. 각 문제는 지배방정식을 설명하고, 지배방정식을이용하여 컴포넌트로 개발하기 위한 코드를 제시한 후, 개발된컴포넌트를 이용하여 시뮬레이션 모델을 구동한 결과를 도시하고, 고참과정을 통하여 편의성을 평가하고자 하였다.

## 4.4.2 연직배수를 고려한 1차원 토질 압밀

일반적으로 흙은 압축성 물질로서, 외부 하중이 가해지면 체적이 감소한다. 이와 같은 현상을 공학적으로 분석하면, 흙은 흙 입자와 물 입자, 공기로 구성되는데, 흙 입자와 물 입자는 비압축성이므로, 흙의 체적 감소는 흙 입자 사이의 간극을 채우고 있는 공기가 압축되거나, 물 입자가 빠져나가는 현상에 기인한다. 만약 흙이 하중을 받기 전부터 수분이 포화되어 있다면, 압축이 일어나기 전에 물 입자가 먼저 빠져나가야 한다. 이때 압축속도는 물이 얼마나 빨리 빠져나갈 수 있느냐에 달려있다.

정리하면, 완전히 포화되어 있거나 또는 부분적으로 포화되어

있는 흙에 하중이 가해지면, 하중으로 인해 간극수압이 발생한다. 이를 과잉간극수압(excess pore water pressure)이라고 하는데, 이 수압으로 인해 임의의 두 점 사이에 수두 차가 발생하면 물이흙 속을 흐르게 된다. 물이 흙 속을 흘러가면, 흐르는 속도는 투수계수에 의존하게 되는데, 예를 들어, 투수계수가 작은 점토에서는 흐름이 매우 느리다. 이와 같이 오랜 시간에 걸쳐 흙속에서 물이 흘러나가면서 흙이 천천히 압축되는 현상을 압밀(consolidation)이라고 한다.

흙에 대한 압밀 거동을 파악한다는 것은 시간과 변위, 즉 압밀 종료시간과 압밀 변형량을 얻고자 함이다. 그러나 흙은 물과 흙 입자로 이루어진 이상(two phase) 구조이며 흙의 압밀은 투수계수, 간극비, 포화도 등 여러 가지 조건에 영향을 받아 이에 대한 해석은 매우 복잡하다. 이러한 복잡한 압밀 거동은 성토나 연약지반 개량문제에 있어서 중요하게 다뤄지고 있다.

#### 1) 토질 압밀의 지배방정식

1차원 압밀이론은 흐름과 변형이 수직방향으로만 일어나는 경우로 토체의 변형을 1차원으로 보고 흙의 압밀 거동을 해석하는 방법이다. Terzaghi는 압밀 과정에서 간극수의 흐름과 점토층의 압축이 수직방향으로만 발생하며 완전포화상태로 가정한 1차원 압밀에 대한 관계식을 과잉간극수압, 토층의 두께, 깊이, 시간과의 관계에 대하여 다음과 같은 방정식을 제시하였다(Terzaghi 1943).

$$c_{v} \frac{\partial^{2} u_{e}}{\partial z^{2}} = \frac{\partial u_{e}}{\partial t} \tag{16}$$

$$c_{v} = \frac{k}{m_{v} \times \gamma_{w}} \tag{17}$$

여기서,  $c_v$ 는 압밀계수,  $u_e$ 는 과잉간극수압, z는 압밀방향 깊이, t는 압밀 시간,  $\gamma_w$ 는 물의 단위중량,  $m_v$ 는 체적변화계수를 의미한다. 이 식은 질량보존법칙을 수용한 이론이며, 점토 지반에서 시간과

깊이에 따라서 과잉간극수압이 어떻게 변하는지에 대한 특성을 규정한 지배방정식이다. 여기서 가정조건은 다음과 같다.

- 1) 흙은 균질하고 완전히 포화되어 있다.
- 2) 흙 입자와 물의 압축성은 무시한다.
- 3) 물의 이동은 Darcy의 법칙을 따르며, 투수계수는 일정하다.
- 4) 압축 토층은 횡적으로 변위되지 못하도록 구속되어 있다.
- 5) 투수계수는 간극비의 변화, 온도에 대해 일정하다.

조건에 따라 압밀방정식을 차분식으로 변환하기 위해 다음과 같이 배수조건은 양면배수로 가정하여 경계조건을 설정하고, 차분식을 전개하였다.

$$U(y, 0) = u_0$$
 :  $t = 0, 0 \le y \le 2H$   
 $U(0, t) = 0, U(2H, t) = 0$  :  $t > 0$  (18)

식 (16)의 좌변항을 정리하면,

$$\frac{\partial^{2} u}{\partial z^{2}} = \frac{1}{2\Delta z} \left[ \frac{\partial U(i+1,j)}{\partial z} - \frac{\partial U(i-1,j)}{\partial z} \right] 
= \frac{1}{2\Delta z} \left[ \frac{U((i+1)+1,j)-U((i+1)-1,j)}{2\Delta z} - \frac{U((i-1)+1,j)-U((i-1)-1,j)}{2\Delta z} \right] 
= \frac{1}{4(\Delta z)^{2}} \left[ U(i+2,j)-U(i,j)-U(i,j)+U(i-2,j) \right]$$
(19)

여기서,  $2\Delta z = \Delta Z$ 라 가정하면 다음과 같이 정리할 수 있다.

$$\frac{\partial^2 u}{\partial z^2} = \frac{1}{(\Delta Z)^2} \left[ U(i-1,j) - 2U(i,j) + U(i+1,j) \right]$$
 (20)

식에서  $\Delta z$ 는 미소 구간을 의미하는 임의 구간이므로,  $\Delta Z$ 를  $\Delta z$ 로 고쳐 쓰면 다음과 같다.

$$\frac{\partial^2 u}{\partial z^2} = \frac{1}{(\Delta z)^2} \left[ U(i-1,j) - 2U(i,j) + U(i+1,j) \right]$$
 (21)

식 (16)의 우변항을 정리하면, 다음 식 (22)와 같다.

$$\frac{\partial u}{\partial t} = \frac{U(i, j+1) - U(i, j)}{\Delta t} \tag{22}$$

정리된 식 (21), (22)를 식 (16)의 좌변항과 우변항에 대입하여 다음과 같이 정리하였다.

$$\frac{U(i,j+1) - U(i,j)}{\Delta t} = c_v \frac{1}{(\Delta z)^2} \left[ U(i-1,j) - 2U(i,j) + U(i+1,j) \right]$$
(23)

여기서  $\Delta t$  와  $\Delta z$  는 각각 시간과 깊이에 대한 미소구간을 나타내므로, 다시 정리하면 다음 식과 같다.

$$U(i, j+1) = rU(i-1, j) + (1-2r)U(i, j) + rU(i+1, j)$$
(24)

$$r = \frac{c_v \times \Delta t}{(\Delta z)^2} \tag{25}$$

$$U(i,j) \equiv U(z,t) \tag{26}$$

$$U(i\pm 1, j+1) \equiv U(z\pm \Delta z, t+\Delta t)$$
(27)

정리한 계산과정은 Figure 35와 같이 도식화하였다(봉태호 et al. 2012). 깊이 방향으로 현상으로 보이는 흙의 과잉간극수압이 배치되며, 횟방향인 시간방향으로 계산과정이 진행됨에 따라 흙

요소가 축차적으로 상호 영향을 미치게 되는 구조로 이해할 수 있다.

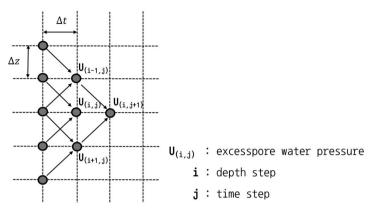


Figure 35 Depth-time grid for simulating consolidation

#### 2) 토질 압밀 컴포넌트의 구현

시스템 시뮬레이션을 위하여 서브시스템으로 단위체적의 간극수압을 모의하도록 컴포넌트를 구현하였다. 계산에 이용되는 변수가 많지 않기 때문에 기본 컴포넌트에서 주요 프로시저를 구현하는 것으로 손쉽게 컴포넌트를 작성하였다.

초기화 과정에서 투수계수, 시간 간격, 위치 간격과 같은 속성을 입력자료로부터 설정하고, 주요 프로시저로 식 (24)에서 정리한 수식을 계산하도록 Figure 36과 같이 흙에 대한 단위 컴포넌트를 작성하였다. 매 계산과정마다 투수율을 업데이트하도록 작성하였는데, 향후 간극비의 변화나 온도에 대해서도 확장 가능할 것으로 생각된다.

```
01:
     @Override
     public void postProcess() {
02:
03:
         setDouble(KEY_U, sumU);
         setDouble(KEY_R, cv * dt / (Math.pow(dz, 2)));
04:
05:
06:
     @Override
     public void process(List<TreeNode> neighborhoods) {
07:
08:
         sumU = 0;
09:
         for (TreeNode n : neighborhoods) {
10:
             IASAPComponent nm = n.getModel();
            sumU += nm.getDouble(KEY_R) * nm.getDouble(KEY_U);
11:
12:
13:
         sumU += (1 - 2 * getDouble(KEY_R)) * getDouble(KEY_U);
```

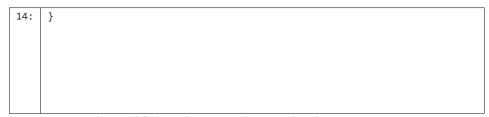


Figure 36 A main code of consolidation component

단위 컴포넌트가 경계요소일 경우에는 별 다른 계산 과정 없이 주어진 경계조건을 전달하도록 구현하였다. 또한 특정 시점의 깊이별 과잉간극수압에 대한 그래프를 그리기 위한 보고용 컴포넌트를 작성하였다. 보고용 컴포넌트는 흙 컴포넌트와 연결되어 있고, 지정한 시점이 되었을 때, 과잉간극수압 값을 수집하여 ASAP 환경으로 전달해주는 기능을 수행한다.

# 3) 연직배수를 고려한 1차원 토질 압밀 모의

흙의 압밀을 모의하기 위하여, 관련 연구(봉태호 et al. 2012)의 초기조건으로 H=3m,  $c_v=0.003$   ${\rm cm}^2/{\rm sec}$ ,  $\Delta t=10,000$ 초,  $\Delta z=10$  m,  $u_0=1$ 를 차용하여 시뮬레이션을 실시하였다. 앞서 구현한 압밀 컴포넌트를 61개 생성하여 초기조건을 입력하고, 특정시점에서 토질 상태를 파악하기 위한 리포트 컴포넌트 4개를 연결하였다. 시스템 환경을 설정하여 ASAP를 구동하고, Figure 37과 같이 모의 결과를 확인하였다.

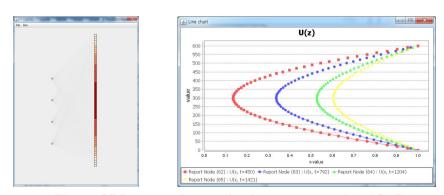


Figure 37 Results of consolidation problem using ASAP

요소의 관계망을 표시한 화면에서는 요소의 과잉간극수압에 따라 요소의 색이 동적으로 바뀌도록 구현하였다. 동적으로 색이 바뀌는 모습을 시간 순서대로 나열하면, 앞서 Figure 35와 같은 그림으로 이해할 수 있다. 관계망에서 왼쪽에 표시된 4개의 노드는 보고용 컴포넌트로 오른쪽 결과 그래프를 그리기 위하여 배치하였다.

시뮬레이션을 구동하는 동안 사용자의 화면 오른편에서 갱신되는 결과 그래프는 특정 시점의 깊이별 과잉간극수압에 대한 분포를 동적으로 도시화한다. 도시한 결과는 50일, 95일, 145일, 160일이 지난 시점의 상태를 나타내며, 매개변수를 참조한 연구(봉태호 et al. 2012)의 결과와 비교하여 동일한 결과를 도출한 것으로 판단된다.

흙의 수직 압밀에 대하여 기존 연구를 재현한 결과, 구현이용이하고, 모의 결과를 직관적으로 제시한 것으로 평가할 수 있다. 구현한 컴포넌트는 독립적으로 수행 가능하므로 도메인을 확장하여풀이할 수 있고, 향후 컴포넌트의 계산식 개량을 통하여 수평방향압밀 및 수직 압밀과 수평 압밀을 결합한 3차원 압밀해석 모형으로확장 가능할 것으로 사료된다.

# 4.4.3 점적 관개에 의한 토양수분 모의

관개 시스템은 농업을 지탱하는 대표적인 시스템으로, 주로 주곡생산을 위하여 논 관개 중심으로 발달되었다. 그러나 최근에는 시설재배가 늘어남에 따라, 정밀관개에 대한 관심이 커지고 있다. 정밀관개는 작물에게 적절한 수량을 공급하여 수분 스트레스를 최소화함으로써, 농작물의 생산량 및 품질 향상을 도모하고, 물소비량 및 양분 용탈, 집적을 감소하고자 하는 방법이다. 정밀관개의 대표적인 기법 중 하나인 점적 관개는 작물의 근군역에관개용수를 직접 공급함으로써 효율적인 관개를 목표로 한다.

점적 관개의 효율성을 높이기 위해서는 작물의 수분 스트레스를 평가하고, 이를 바탕으로 적절한 관개량 및 시기를 결정해야 한다. 수분 스트레스는 근군역에 분포하는 토양수분에 따라 평가할 수 있으므로, 최우선적으로 토양수분에 대한 모니터링이 필요하다. 실험적 모니터링이 가장 좋은 방법이지만, 조사 비용 및 시간의 문제, 제한된 조사 지점 수와 같은 한계로 수치해석적 모니터링이 필요하다.

본 절에서는 점적관개에 의한 토양수분을 모의하기 위하여 토양 내 수분이동 컴포넌트를 개발하고, ASAP 환경에서 2차원 토양수분 변화를 모의하였다.

### 1) 토양 수분 거동의 지배방정식

다공성 물질내의 유체흐름은 유량이 비교적 적을 때, 유량이 손실수두와 비례한다는 1856년 Darcy의 실험을 통하여, 다음과 같은 식을 소개되었다(윤용남 2007).

$$Q = kA \frac{h_1 - h_2}{ds} \tag{28}$$

여기서, k는 비례상수로서 수리전도도 (ms), A는 다공성 물질의 단면적  $(m^i)$ ,  $h_i$ 은 유입수의 총 수두,  $h_2$ 는 유출수의 총 수두, ds는 흐름의 길이 (m)를 의미한다. 수리 경사 i는 에너지 손실율로서, 단위 길이당 수두 변화량으로 식 (29)와 같이 정의되며, 수리 경사를 이용하여 식 (28)를 다시 정리하면, 유량은 식 (30)과 같으며, 이는 통수면적에 유속을 곱한 값과 일치해야 한다.

$$i = \frac{h_1 - h_2}{ds} = -\frac{dh}{ds} \tag{29}$$

$$Q = kAi = VA \tag{30}$$

따라서, Darcy 방정식은 다음과 같이 정리된다(윤용남 2007).

$$V = ki = -k\frac{dh}{ds} \tag{31}$$

인위적으로 교란된 소규모 영역에서 수리전도도의 방향성은 측정이 매우 어려울 뿐 아니라 무시될 수 있는 정도이기 때문에(노희명 et al. 2000), 토양이 균질하여 각 방향의 투수계수가 동일하다고 가정하면 다음과 같이 나타낼 수 있다.

$$\vec{V} = -k\nabla h = -k\nabla \left(\frac{p}{\gamma} + z\right) \tag{32}$$

불포화토양에서의 물은 인장응력을 받고 있으므로 음의 수압이 작용한다고 할 수 있으며, 이를 토양수분 장력(matric head) → 로 나타낸다. 기준면을 지표면으로 하면 지표면 아래로 위치수두가 → 이므로 불포화토양에서의 Darcy 방정식은 식 (31)을 수정하여 다음과 같이 표현된다.

$$\vec{V} = k\nabla(\psi + z) \tag{33}$$

토양에서의 수분의 이동은 연속적으로 일어나며, 이러한 시스템에서 수분이동을 정량적으로 기술하기 위하여 물질보전 원리인 물수지 방정식을 통제용적에 대하여 적용하면 다음 식과 같다.

$$\frac{\partial}{\partial t} \iiint \theta dV = \iint_{S1} \vec{q} \ \vec{n} \ dS - \iint_{S2} \vec{q} \ \vec{n} \ dS \tag{34}$$

여기서, V는 통제용적의 부피( $L^2$ ),  $\theta$ 는 함수율( $L^2/L^3$ ), q는 Darcy의 속도( $L^2/T/L^2$ ), S는 통제용적의 표면, S1은 유입부의 표면, S2는 유출부 표면, n은 통제용적 표면에서 바깥쪽으로 수직인 단위 벡터를 나타낸다. 식 (34)의 좌변은 통제용적에 물이 축적되는 속도, 우변의 첫째 항은 통제용적에 들어오는 물의 유입을, 둘째 항은 통제 용적에서 나가는 물의 유출율을 각각 나타낸다. 이두 항은 같은 현상이 반대 방향으로 일어나는 것에 불과하므로 이를 통합하여 Divergence theorem을 적용하면 다음과 같이 식을 정리할 수 있다.

$$\iint_{S1} \vec{q} \ \vec{n} \ dS - \iint_{S2} \vec{q} \ \vec{n} \ dS = -\iint_{S} \vec{q} \ \vec{n} \ dS = \iiint_{V} \nabla \ \vec{q} \ dV$$
(35)

식 (34)와 (35)를 등치시키면, 식 (36)과 같고, 이로부터 물수지의 일반적인 형태를 구하여 식을 도출할 수 있다.

$$\iiint_{V} \frac{\partial \theta}{\partial t} dV + \iiint_{V} \nabla \vec{q} \ dV = 0$$
 (36)

$$\frac{\partial \theta}{\partial t} + \nabla \vec{q} = 0 \tag{37}$$

식 (37)을 식 (33)에 대입하여 정리하면 다음과 같다.

$$\frac{\partial \theta}{\partial t} + \nabla (K\nabla \psi) + \nabla (K\nabla z) = 0 \tag{38}$$

일반적으로 불균질한 흙에서 수리전도도 K 는 위치에 따라 방향성이 변하는 텐서이나, 인위적으로 교란된 소규모의 영역에서 그 방향성은 측정이 매우 어려울 뿐 아니라 무시될 수 있는 정도이기 때문에 스칼라로 취급한다. 또한, 함수율  $\theta$  는 matric potential  $\forall$  만의 함수로 나타낼 수 있으므로, 식 (39)를 적용하여, 식 (40)과 같이 쓸 수 있다.

$$\frac{\partial \theta}{\partial t} = \frac{\partial \theta}{\partial \psi} \frac{\partial \psi}{\partial t} \tag{39}$$

$$\frac{\partial \theta}{\partial \psi} \frac{\partial \psi}{\partial t} + \nabla K \nabla \psi + K \nabla \nabla \psi + \nabla K \nabla z + K \nabla \nabla z = 0 \tag{40}$$

여기에 2차원 직교좌표계를 적용하고, 관개량 Q를 외부에 의한수분 유입량으로 대입하면, 다음과 같이 mixed-form Richards equation인 식 (41)로 표현된다(Celia et al. 1990).

$$\frac{\partial \theta}{\partial \psi} \frac{\partial \psi}{\partial t} = \frac{\partial}{\partial x} \left[ K \frac{\partial \psi}{\partial x} \right] + \frac{\partial}{\partial z} \left[ K \left( \frac{\partial \psi}{\partial z} + 1 \right) \right] + Q \tag{41}$$

Richards equation은 토양의 용적수분함량에 따라서 수분포텐셜과 불포화수리전도도가 변하므로, 이에 대한 관계식이 필요한데, 본절에서는 식 (44)와 같이 van Genuchten의 실험에 의한 관계식을 이용하였다(van Genuchten 1980).

$$\psi = \frac{1}{\alpha} \left\{ \left[ \frac{\theta_s - \theta_r}{\theta - \theta_r} \right]^{\frac{1}{M}} - 1 \right\}^{\frac{1}{N}}$$
(42)

$$Se = \frac{1}{\left[1 + (\alpha \psi)^{N}\right]^{M}} \tag{43}$$

$$K(\psi) = \frac{K_s \left\{ 1 - (\alpha \psi)^{mn} \left[ 1 + (\alpha \psi)^n \right]^{-m} \right\}^2}{\left[ 1 + (\alpha \psi)^n \right]^{ml}}$$
(44)

여기서,  $K_s$ 는 포화 수리 전도도 (cm/s), m=1-1/n, l은 0.5를 사용하였다(Mualem 1976). 식 (41)을 풀이하기 위하여 유한차분식으로 전개하면, 다음과 같이 정리된다.

$$\theta_{i,k}^{n+1} = \theta_{i,k}^{n} - \frac{\Delta t}{(\Delta x)^{2}} \begin{cases} K_{i+1/2,k}^{n} (\psi_{i+1,k}^{n} - \psi_{i,k}^{n}) \\ + K_{i-1/2,k}^{n} (\psi_{i-1,k}^{n} - \psi_{i,k}^{n}) \end{cases}$$

$$- \frac{\Delta t}{(\Delta z)^{2}} \begin{cases} K_{i,k+1/2}^{n} (\psi_{i,k+1}^{n} - \psi_{i,k}^{n} + \Delta z) \\ + K_{i,k-1/2}^{n} (\psi_{i,k-1}^{n} - \psi_{i,k}^{n} - \Delta z) \end{cases} + \Delta t Q$$

$$(45)$$

여기서, i, k는 각각 x, z 방향으로의 유한차분 격자의 좌표를 의미하며, Q는 단위시간당 관개량을 의미한다. 수리전도도는 다음 식과 같이 인접한 컴포넌트 사이의 값을 적용하였다(Sadegh Zadeh

2011).

$$K_{i\pm 1/2} = \frac{K_i + K_{i\pm 1}}{2} \tag{46}$$

# 2) 토양 수분 컴포넌트의 구현

토양 수분의 재분배 모의를 위하여, 앞서 정리한 Richards 방정식을 Figure 38과 같이 작성하여, 토양 수분 컴포넌트를 구현하였다. 컴포넌트의 주요 프로그램 코드를 살펴보면, 준비작업, 정보교환작업, 후속작업으로 구분하여 각각 인터페이스에서 제공하는 메소드를 오버라이드하여 작성하였다. 준비작업에서 계산되는 수리전도도는 가독성을 위하여 따로 메소드로 추출하였다.

```
01:
     private void vanGenuchten() {
02:
         double M = 1 - 1/get(N);
03:
         //--- soil water content -> soil water potential ----//
04:
         set(Psi, (1 / get(A)) * Math.pow(Math.pow(
                (get(ThS) - get(ThR)) / (get(Th) - get(ThR)), 1 / M) - 1,
05:
06:
                1 / get(N)));
         //--- soil water potential -> hydraulic conductivity ----//
97.
08:
         double Se = Math.pow(1/(1+Math.pow(get(A)*get(Psi),get(N))),M);
09:
         setDouble("Se", Se);
10:
         set(K, get(Ks) * Math.pow(Se, 0.5) * Math.pow(1
11:
                 -Math.pow(get(A) * get(Psi), get(N) - 1) * Se, 2));
12:
13:
14.
     @Override
15:
     public void preProcess() {
         vanGenuchten();
16:
17:
18:
19:
     @Override
20:
     public void process(List<TreeNode> neighborhoods) {
21:
         double Th2temp = get(Th);
22:
23.
         for (TreeNode n : neighborhoods) {
24:
             IASAPComponent nm = n.getModel();
25:
             double kValue = (get(nm, K) + get(K)) / 2;
26:
             double DIR = 0;
27:
             if (get(nm, y) - get(y) > 0.01) {
                DIR = 1;
28:
29:
             } else if (get(nm, y) - get(y) < -0.01) {
30:
                DIR = -1;
             }
31:
             Th2temp += (get(dT)/Math.pow(get(dX),2))
32:
33:
                       *(kValue*(-(get(nm, Psi)-get(Psi))+DIR*get(dZ)));
34:
35:
         setDouble(Th2, Th2temp);
36:
     }
37:
38:
     @Override
39:
     public void postProcess() {
         setDouble(Th, Th2);
40:
41:
```

Figure 38 A main code of distribution component for soil water

작성된 코드를 살펴보면, 준비작업에서는 van Genuchten식을 통해 각 컴포넌트의 수리전도도를 갱신한다. 정보교환작업은 인접한 컴포넌트에서 정보를 획득하여 토양수분을 계산하는 과정을 기술한 코드로서, 19-37라인에서 확인할 수 있다. 특히 z축 방향으로는 계산항이 추가되므로, 방향을 의미하는 DIR 변수를 정의하였다. 마지막으로 후속작업에서는 임시변수인 Th2에 저장되어 있는 토양수분 값을 이용하여, 현재 시점의 토양수분으로 갱신한다.

컴포넌트 구현에 있어서 정보교환작업 과정에서 정보를 바로

갱신하지 않고, 주변 컴포넌트에서 필요한 정보를 처리한 이후, 정리 작업시점에서 정보를 갱신하도록 컴포넌트를 구현하였다. 이는 모든 컴포넌트가 서로 정보를 교환하는 시점에 동일한 변수를 참조하도록 변수의 불변성을 유지하기 위함이다. 데이터의 불변성을 확보하면, 데이터를 다루는 입장에서 복잡도가 감소하고, 프로그램 작성 시 오류를 줄일 수 있다(Bloch 2008).

### 3) 점적 관개에 의한 토양수분 모의

점적 관개에 따른 토양수분을 모의하기 위하여, 깊이  $60 \ cm$ , 폭 $100 \ cm$ 으로 도메인을 설정하고, 격자 간격은  $5 \ cm$ 로 컴포넌트를 연결하였다. 토양층은 Table 6에서 정리한 토성별 속성 (최순군 et al. 2012)을 기준으로 상부  $30 \ cm$ 는 sandy loam으로 설정하고, 하부  $30 \ cm$ 는 loam으로 컴포넌트의 속성을 입력하였다.

점적 호스를 이용하여 폭 방향의 중앙 지점에서 관개를 4시간 지속하고, 이후 48시간동안 토양수분 변화를 모의하였다. 관개수량은 2.25 l/hr로 가정하고, 시간 간격은 60초 간격으로 설정하여, 시뮬레이션을 수행하였다.

Table 6 Soil hydraulic parameters for textural class (Choi et al., 2012)

Textural	$\theta_r$	$ heta_{\scriptscriptstyle S}$	α	n	$K_s$
class	$(m^3/m^3)$	$(m^3/m^3)$	$(cm^{-1})$	_	(cm/day)
Sandy loam	0.039	0.387	0.027	1.45	38.20
Loam	0.061	0.399	0.011	1.47	12.00

관개가 끝난 4시간 후와 토양수분 재분배를 살펴보기 위하여 12시간 후, 24시간 후, 48시간 후를 모의하여 결과를 Figure 39와 같이 도시하였다. 관개가 끝난 직후에는 근군역 구간에 높은 수분율을 보이나 시간이 지나면서 점차 수분이 주변지역으로 재분배되는 현상을 확인할 수 있었다.

특히 상부 30 cm 토층은 투수율이 높은 sany loam으로 설정하고, 하부 30 cm 는 상대적으로 투수율이 낮은 loam으로 속성을 설정하였기 때문에, 경계층에서 수분이 정체하는 모습을

확인할 수 있었다. 시뮬레이션 결과는 실험 결과와 정량적으로 비교할 수는 없었으나, 토양층 특성과 관개수량이 유사한 실험연구의 결과물(최순군 et al. 2012)과 비교할 때, 유사한 패턴을 확인할 수 있었다.

본 절에서는 2차원 토양 수분의 재분배 컴포넌트를 개발하여, 토양수분의 분배과정을 모의하였다. 컴포넌트를 이용하여 시뮬레이션을 수행할 때는 관개량, 관개 시간, 토양 특성과 같은 외부조건만 입력하고, 컴포넌트를 연결함으로써 문제 구성을 할 수 있다. 구성된 모형은 시뮬레이션 구동과 함께 관심있는 속성에 대한 실시간 변화량을 관찰할 수 있고, 모델에서 수분량을 회색의 농도로 표시하도록 설계하여, 별다른 추가 작업없이 수분분포를 정성적으로 파악할 수 있다. 따라서 프로그래밍에 익숙하지 않은 연구자도 간편하게 시뮬레이션을 구동할 수 있다.

본 연구의 모형을 이용하여 점적 관개를 위한 간격 설정이나 관개 시간, 관개량 등의 평가가 가능할 것으로 판단된다. 향후 작물 컴포넌트를 개발하여 결합하면, 작물 소비량을 고려한 토양수분 재분배 모형으로 발전시킬 수 있으며, 토양 수분을 고려한 작물 성장 모형이나, 수분 스트레스 모형 등이 가능할 것으로 사료된다.

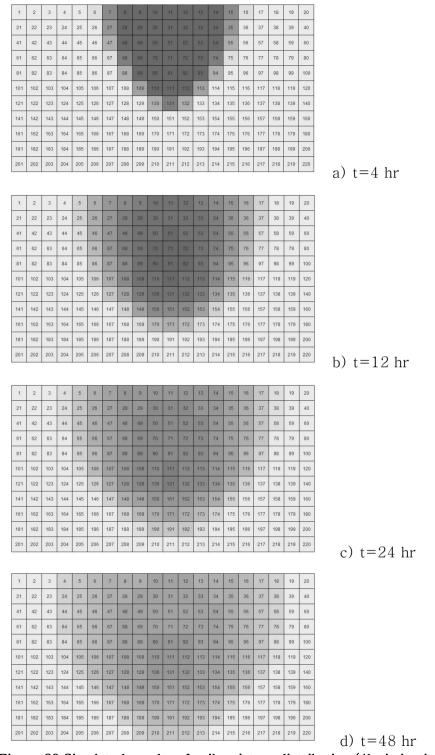


Figure 39 Simulated results of soil moisture distribution (4hr irrigation)

## 4.4.4 배수갑문 염해 모의

해양에 시공된 철근콘크리트 구조물은 염해, 중성화, 동결융해 와 같은 다양한 노후화 조건에 노출되어 있다. 이러한 시설물은 그 규모와 설치 위치로 인해 기능을 다하지 못할 경우 사회적, 환경적, 경제적으로 피해 규모가 크다. 그럼에도 불구하고, 조수와 파랑을 항상 견뎌야 하는 구조물의 특성상 구조물의 상태 파악 및 보수, 보강이 어려워 관리가 채계적으로 이루어지지 못하고 있다. 따라서설계 및 시공 단계에서부터 운영 및 폐기에 이르기까지 효율적인 관리를 위하여 시설물의 노후화에 대한 지속적인 모니터링이 필요하다. 시뮬레이션은 효과적인 모니터링 도구로서, 해양 구조물 중 철근콘크리트로 시공된 배수갑문 피어의 노후화를 계산함으로써 시설물의 상태를 효과적으로 예측할 수 있다.

해양 철근콘크리트 구조물의 노후화의 주된 요인인 염해는 확산방정식을 이용하여 해석할 수 있는 것으로 알려져 있다. 확산 방정식을 일반적으로 수치해석을 통해 해결 가능하나, 방정식을 구성하는 조건이 계산 중간에 바뀔 경우에는 해석이 다소 복잡해진다. 구조물의 내부적으로는 염분이온의 확산속도가콘크리트의 노후화 정도에 따라 달라지게 되고, 외부적으로는 보수,보강을 통해서 콘크리트 구조물 일부 구간의 물성치를 인위적으로변화시키기도 한다. 따라서 확산방정식을 구성하는 확산계수가상이해지므로, 연속방정식의 형태로 문제를 표현하기가 어려워진다. 더욱이 중성화나 동결융해를 고려하면, 염분의 확산속도에 관한함수가 다변수가 되어, 연속방정식을 수립하기가 더 어려워진다는 문제점이 발생한다.

구조물을 구성하는 작은 부분을 독립된 서브시스템으로 가정하고, 작은 서브시스템이 모여서 구조물의 거동을 설명할 수 있다고 가정하면, 각 서브시스템은 주변 환경조건에 따라 상태를 변화시키는 구조로 설명할 수 있다. 따라서 도메인영역 전체를 설명하는 연속방정식을 수립하지 않고, 일부 도메인에서 거동을 설명할 수 있는 ASAP 컴포넌트를 개발하여, 이를 연결함으로써 시스템을 모의할 수 있다.

#### 1) 콘크리트 내 염분 이동방정식

해양 콘크리트 구조물 외측면에 존재하는 해수의 염분이 콘크리트 내부로 침투해 들어가는 것은 Fick의 제2법칙으로 표현할 수 있음이 알려져 있다.

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} \tag{47}$$

여기에서 D는 염소이온 확산계수( $cm^2/\sec$ ), t는 시간( $\sec$ ), x는 거리(cm)를 의미한다. 이 식을 통해 해당 위치에서 시간의 흐름에 따른 염분량의 변화를 계산할 수 있다. 위의 식은 2차원 양함수 유한차분법으로 전개하여 아래 식과 같이 나타낼 수 있는데, 이식은 t+1 시점에서 이웃한 컴포넌트의 t 시점에서의 속성 값을 이용하여 해당 컴포넌트의 염분량을 계산하는 식이라고 볼 수 있다.

$$C_{m,n}^{t+1} = F_o \left( C_{m+1,n}^n + C_{m-1,n}^t + C_{m,n+1}^t + C_{m,n-1}^t \right) + \left( 1 - 4F_o \right) C_{m,n} \tag{48}$$

이 때  $F_o$ 는 아래 식 (49)로 표현할 수 있으며, 그 값이 1/4보다 작은 경우에만 해에 수렴한다는 특성을 갖고 있다.

$$F_o = \frac{D\Delta t}{\left(\Delta x\right)^2} \tag{49}$$

여기서 D는 식 (47)의 D와 동일한 값인데, 콘크리트 재료의 특성, 중성화나 동결융해 여부 등에 따라 달라지며, 시간이 흐름에 따라 콘크리트의 연령에 대해서도 계속 변화하는 값이다. 이에 대해 Maage는 1996년에 아래 식과 같이 정리하였다(Maage et al. 1996).

$$D(t) = D_{ref} \cdot \left(\frac{t_{ref}}{t}\right)^m \tag{50}$$

여기서, D(t) 는 시간 t 에서의 확산계수( $cm^2/\sec$ )이고,  $D_{ref}$  는 기준시간에서의 염소이온확산계수( $cm^2/\sec$ ),  $t_{ref}$  는 기준시간으로, 본 절에서는 12개월을 의미한다. t 는 경과시간, m 은 확산계수 감소 상수를 나타낸다. 따라서 실험을 통해  $t_{ref}$  시점에서의 염소이온확산계수  $D_{ref}$  를 구하면, 시간이 흐름에 따라 감소하는 확산계수를 추정할 수 있다.

# 2) 콘크리트 내 염분확산 컴포넌트의 구현

콘크리트 내 염분확산 컴포넌트는 식 (48)-(50)을 이용하여 염분을 계산하도록 구현하였다. 각 컴포넌트는 아래 Figure 40의 분홍색 요소와 같이 하나의 컴포넌트가 가로, 세로 모두 dx의 길이를 갖는 정사각형 영역을 대표하는 것으로 표현하였으며, 각각의 컴포넌트가 dx와 dt의 값을 저장하고 있도록 설정하였다. 또한 염분의 확산계수는 해당 컴포넌트가 대표하고 있는 콘크리트의 성질과 연령에 따라 영향을 받으므로 매 계산 시간마다 확산계수가 재계산되도록 구현하였다. 이렇게 계산된 확산계수는 dt와 dx와의 계산을 통해 식 (49)와 같이  $F_o$ 를 산정하였으며, 이 값이 수렴반경 1/4를 초과할 경우 사용자가 재설정 할 수 있도록 프로그램이 정지하도록 개발하였다. 개발한 컴포넌트의 주요코드는 Figure 41에 도시하였으며, 이를 통해 주변 컴포넌트의 이전 계산시간 염분량을 이용해 해당 컴포넌트의 현재 시간 염분량을 계산할수 있다.

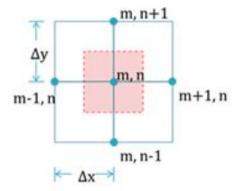


Figure 40 Explicit finite difference method

```
@Override
01:
02:
     public void process(List<TreeNode> neighborhoods) {
03:
         double dt = this.getDouble("dt");
         double dx = this.getDouble("dx");
04:
05:
06:
         String material = this.getString("material");
07:
08:
         double diffusionCoefficient =
             DiffusionCoefficient.chloride(material, this.age);
09:
10:
11:
         double Fo = diffusionCoefficient * dt / (dx * dx);
12:
13:
         if(Fo > 0.25){
14:
             System.err.printf("Too Big Fo: %f, id: %d%n",
15:
                     Fo, this.getId());
16:
            System.exit(1);
         }
17:
18:
         double sumNeighborTemp = 0.0;
19:
20:
         double preTemp = getDouble("preChloride");
21:
         for (TreeNode neighbor : neighborhoods) {
22:
23:
             double neighborhoodTemp =
24:
                 neighbor.getModel().getDouble("preChloride");
25:
26:
             sumNeighborTemp += neighborhoodTemp;
27:
         }
28:
         double chloride = Fo*sumNeighborTemp + (1-4*Fo)*preTemp;
29:
30:
31:
         this.setDouble("chloride", chloride);
32:
33:
     }
```

Figure 41 A main code of chloride penetration component

#### 3) 배수갑문의 염해모의

해양에 설치되어 염해를 받고 있는 철근콘크리트 구조물을 모의하기 위해 가로와 세로 모두 50cm인 정사각형 기둥을 가정하고, 격자의 크기는 가로, 세로 1cm로 설정하였다. 콘크리트 재료는 SRC-31.4 시멘트를 가정하였으며, 이 재료의 재령 12개월의 염화물 확산계수는  $4.85 \times 10^{-8} cm^2/s$  으로 식 (50)에 도입하여 매시간별 염화물확산계수를 산정하였다.

콘크리트 구조물 표면은 해수에 맞닿게 되므로 일본 청수항 방파제에서 장기 폭로실험을 통해 얻은 수치를 이용하여 0.346%로 설정하였다. 철근은 표면으로부터 10cm 깊이에 위치하고 있는 것으로 가정하고, 이 지점에서의 염분량 증가를 시간에 따라 분석하였다. 여기에서는 보수 및 복구 없이 염해만 발생하는 단순염해 조건과 20년에 한번씩 콘크리트 구조물의 표면을 걷어내고 새로운 콘크리트를 타설하는 보수공법을 모의하였다. 개발한 컴포넌트를 이용하여 ASAP 상에서 시뮬레이션을 구동하는 화면은 Figure 42와 같다.

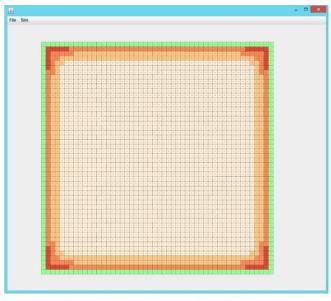


Figure 42 Modelling for chloride attack of marine concrete structure

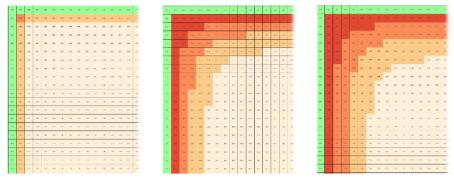


Figure 43 Chloride attack progress on the corner at 1st year, 10 years later, and 20 years later

사각형 콘크리트 기둥 단면의 한 모서리를 Figure 43과 같이 도시하였다. 각 그림에서 구조물 준공 후 1년, 10년, 20년이 경과함에 따라 염분이 콘크리트 내부로 침투한 정도를 확인하였다.

표면으로부터 10cm 깊이의 철근 위치에서 염화이온량의 변화량을 Figure 44와 같이 그래프로 도시하였다. 철골 철근콘크리트인 SRC-31.4의 임계허용 염화이온량은 0.027%로 알려져 있으며, 시공 후 약 32년이 경과한 시점에서 임계허용 수치에 도달하는 것으로 나타났다.

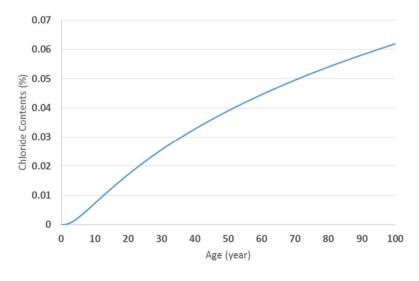


Figure 44 Chloride attack progress at a 10cm deep point from surface without repair or recover treatment

### 4) 배수간문의 복구 시 염해모의

해양 콘크리트 구조물의 사용연한을 증가시키기 위한 관리 방법에는 염화물 침투가 일어난 일정 깊이의 콘크리트를 제거하고 새로운 콘크리트로 대체하여, 본래의 형태로 복원하는 복구 방법과 표면의 염화물 침투 속도를 저감시키기 위해 표면을 피복하는 보수의 공법이 있다. 여기에서는 복구 공법에 대해 모의를 하였다.

복구조건을 모의하기 위해서는 사용자가 보수 시점과 보수 깊이를 입력하면, 입력한 값을 기준으로 대상이 되는 요소들의 염화물 이온량을 '0'으로 재설정하고, 확산계수를 계산하는 데에 필요한 연령도 0으로 재설정하도록 컴포넌트를 작성하였다.

본 절에서는 시공 후 20년에 복구 공사를 실시하는 것으로 가정하였으며, 복구공사는 표면으로부터 5cm를 제거하고, 새로운 콘크리트로 시공하는 대체 공사로 가정하였다. 이러한 가정에 따라 시뮬레이션을 수행한 결과, Figure 45에 도시한 바와 같은 결과를 구하였다. 분석결과는 복구 직전인 19년, 복구 시점인 20년, 그리고 복구 후 1년이 지난 21년의 콘크리트 염분 농도를 도시하였다. 염해는 외곽에서부터 염화농도가 커지며, 구석은 중첩효과로 인하여 깊이가 좀 더 깊은 것을 확인할 수 있다. 복구가 실시되면서, 대부분의 콘크리트가 초기조건으로 대치되었음을 확인할 수 있으며, 시공 후 1년이 지난 시점에 5cm 깊이에서 염화농도가 낮아진 모습을 확인할 수 있다. 이는 새로운 콘크리트가 타설되면서 이미 침투한 염화물이 주변의 새로운 콘크리트도 옮겨가면서 염화농도가 낮아진 결과로 풀이된다.

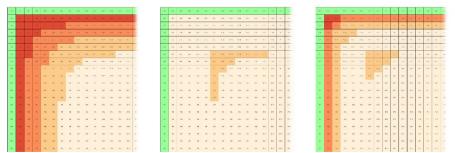


Figure 45 Chloride attack and repair progress at 19 years later, 20 years later, and 21 years later

시공 후 매 20년마다 표면을 복구 공사를 수행하는 경우, 철근 위치인 표면으로부터 10cm 지점에서의 염화물 이온량은 Figure 46과 같이 도시하였다. 실선은 복구 처리를 하지 않은 상태로 염해를 모의한 결과이며, 점선으로 표시된 선은 20년마다 복구시공을 하는 경우를 모의한 결과로 아무런 처리를 하지 않을 때에 비해 느린 속도로 염화이온이 침투하고 있음을 확인할 수 있다.

앞서 철골 철근콘크리트 SRC-31.4의 임계허용 기준치를 기준으로 내구연한을 평가하면, 32년이던 내구연한은 20년에 보수가 이루어진 후, 38년으로 내구연한이 약 6년이 증가하는 것으로 분석되었다.

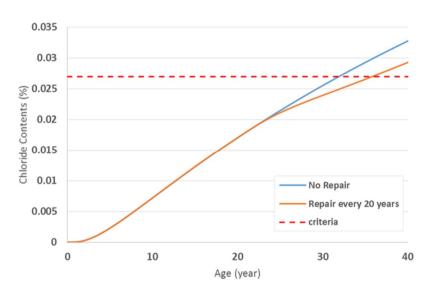


Figure 46 Chloride attack progress considering repair every 20 years at a 10cm deep point from surface without repair or recover treatment

이처럼 사용자는 콘크리트의 재료와 형상을 입력하고, 복구 공사 시점만 입력하면 본 연구에서 제안한 모델에 따라 해양구조물 안전성을 평가할 수 있다. 다양한 시나리오를 적용하여 결과를 비교함으로써, 최적의 보수 계획을 수립할 수 있을 것으로 생각된다. 향후 중성화, 동결융해 등 염해의 진행속도에 영향을 주는 인자들을 고려하여 컴포넌트를 확장할 경우, 보다 정확한 모의 결과를 얻을 수 있을 것으로 판단된다.

# 4.5 이질소통을 하는 위상구조

## 4.5.1 문제 선정

이질소통은 앞서 기술한 동질소통구조와는 달리 연결된 요소들 간의 연결 강도가 다른 문제로 정의할 수 있다. 이질소통 문제는 네트워크 이론에서 가중네트워크로 표현되는 문제로서, 시스템을 구성하는 요소의 속성 못지 않게 요소를 연결하는 연결속성에 따라 시스템의 메커니즘이 영향을 받는다. 전염병은 매개체를 통하여 전염되며 노출빈도가 높을수록 감염율이 높다. 또한, 감염 매개체와의 노출은 좁은 공간에서는 거리에 반비례하지만, 넓은 스케일에서는 공간적 인접성보다는 관계망에 따른 노출 기회의 증가에 더 큰 영향을 받는다. 따라서 본 절에서는 이질소통을 하는 위상구조 문제로 전염병 확산 모형을 적용 예로 선정하였다.

# 4.5.2 전염확률을 고려한 감염농가 확산

고병원성 조류인플루엔자(HPAI, Highly Pathogenic Avian Influenza)는 폐사율이 높고 전염성이 강해서 축산 농가에 큰 피해를 준 사례가 있으나, 근본적인 예방백신이 없기 때문에 발병시인근 농가에 초기 유입 방지 및 확산 방지가 중요하다. 닭은 잠복기가 3일에서 14일로 알려져 있으며, 호흡기를 통하여 동물은 물론, 사람들의 이동을 통해서 전파가 가능하다.

감염농가를 군집화하여 분석한 연구에 따르면, 동일 지역 내에서는 사료차량, 약품 차량과 같은 짧은 거리를 순회하는 기계적 전파가 주요한 감염 원인으로 지목하였으며, 다른 지역 간에는 부화장 차량, 부산물 차량 등이 매체가 되어 감염된 것으로 보고하였다(윤하정 et al. 2008). 다른 지역 간의 전염은 차량 차단 및 주요 운송경로에 방역소를 설치하는 방식으로 대안을 마련하고 있다. 그러나 지역 내 전염은 바이러스의 잠복기, 농장의 늦은 신고, HPAI 감염 여부 판단의 전문성 등으로 초기 감지가 어려워, 현실적으로 살처분 이외에 다른 대안이 없으며, 살처분 역시 일정 반경이내는 일괄적으로 시행하기 때문에, 경제적, 사회적 비용이

매우 크다. 이를 방지하기 위해서는 요인별 위험도 평가 및 발병경로를 추적하여 확산 경로를 파악하는 것이 중요하다.

본 절에서는 2008년 김제시에서 발병했던 HPAI 사례에 대한 역학조사 결과를 바탕으로 농가를 구성하고, 농가간 전염 영향도를 평가하여, 적절한 방제가 이루어지지 않았을 경우의 감염 농가 확산을 모의하고자 하였다.

## 1) 전염확률을 고려한 감염농가의 확산방정식

농가 사이의 HPAI 확산을 모의하기 위하여 농가를 분석단위로 설정하고, 컴포넌트를 구성하였다. 농가의 감염을 평가하기 위하여 감염율 함수를 도입하고, 다음과 같은 상태방정식을 정립하였다.

$$\begin{cases} S(N,t) = \text{clean} &: IR(N,t) < 0.2\\ S(N,t) = \text{infect} &: IR(N,t) \ge 0.2\\ S(N,t) = \text{killed} &: IR(N,t) \ge 0.5 \end{cases}$$
(51)

여기서, S(N,t)는 N 번째 농가의 t시간의 상태를 의미하며, 상태는 청정, 감염, 살처분 상태로 구분된다. IR(N,t)는 잠재 감염율로 N 번째 농가가 t시점에 감염에 대한 기대치로 정의하였다.

잠재 감염율은 농가 내부 요인과 농가 외부 요인으로 구분할수 있으며, 농가 내부 요인은 대상 지역에 HPAI가 발생한 경우, 지역 내 농가의 잠재 감염율은 매 단위시간마다 1%씩 증가하는 것으로 정의하였다. 농가 외부 요인은 인근 농가가 감염이 된 경우, 농가간 전염 영향력을 고려하여, 인근 농가의 감염율이 누적되는 것으로 설계하였다. 만약, 농가가 감염이 확인되어 살처분을 통한 방역이 이루어지면, 살처분 상태의 농가는 주변 농가에 더 이상 영향을 미치지 않는 것으로 설정하였다. 이와 같은 거동에 맞추어 아래 식 (52)와 같이 잠재 감염율에 대한 차분방정식을 수립하였다.

$$IR(N,t) = IR(N,t-1) \times (1+0.01) + IR(M,t-1) \times SR \times R(N,M)$$
 (52)

여기서, N은 대상 농가, M은 인근 농가, SR은 농가간 확산율,

R(N,M) 은 N 농가가 M 농가로부터 감염영향을 받을 확률이며, 만약 M 농가가 청정상태이거나 살처분 상태이면, R(N,M) 이 포함된 계산항은 0으로 가정하였다.

#### 2) 감염농가 컴포넌트의 구현

감염 농가 컴포넌트는 Figure 47과 같이 컴포넌트 프로시저를 작성하였다. 전처리 과정에서는 감염확률이 20% 이상일 경우, 감염상태로 표시하도록 설정하였으며, 50% 이상일 경우에는 살처분처리로 설정하였다. 상태를 저장하면서, 컴포넌트의 색채도 같이지정하여, 네트워크 뷰어에서 시뮬레이션이 지남에 따라 감염정도를확인할 수 있도록 구현하였다. 메인 프로시저에서는 주변 감열율을획득하여, 확산율과 이질소통구조를 갖는 속성인 전염확률 가중치 "ew"를 취득하여, 식 (52)를 구현하였다.

```
01:
     @Override
02:
     public void preProcess() {
03:
         if (infection >= .2 && getString("status").equals("clean")) {
04:
               setString("status", "Infect");
05:
              setString("color", "996666");
06:
         if (infection >= .5) {
07:
              setString("color", "cccccc");
08:
              setString("status", "Kill");
09:
10:
              isComplete_ = true;
11:
12:
         setDouble("infection", infection_);
13:
     }
14:
15:
     @Override
16:
     public void process(List<TreeNode> neighborhoods) {
17:
         double add = 0;
         for (TreeNode n : neighborhoods) {
18:
19:
            IASAPComponent nm = n.getModel();
20:
            if (n.getModel().getString("status") != null
21:
              && n.getModel().getString("status").equals("Infect")) {
22:
                 add += getDouble("ew", n.getModel().getId())
23:
                          *n.getModel().getDouble(("Infect"))
24:
                           *getDouble("spreadRatio");
25:
            }
26:
         infection_ = infection_ * (1+getDouble("incRatio")) + add;
27:
28:
     }
```

Figure 47 A main code of infection component

### 3) 농가 간 전염확률을 고려한 확산모의

구축한 감염농가 확산 컴포넌트를 이용하여 Figure 48과 같이 시뮬레이션 환경을 구성하였다. 농가 위치는 위경도 좌표를 이용하여 표시하였으며, 총 농가 수는 39개로 역학보고서를 참고하여 구성하였다(농림수산식품부 et al. 2008). 동그라미로 표시된 컴포넌트는 각각이 농가를 의미하며, 1번부터 8번 농가는 감염이 확인된 농가이며, 9번부터 13번 농가는 감염이 의심되는 농가이나, 본 연구에서는 동일하게 감염 일자에 감염이 된 것으로 가정하였다.

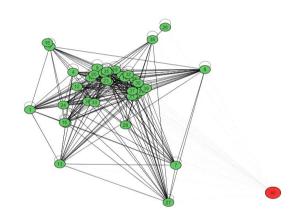


Figure 48 Initial condition of network model for HPAI infection

화면구성을 살펴보면, 오른쪽 아래, 붉은 원은 전체 감염 농가수를 계측하기 위하여 모니터링 컴포넌트이며, 다른 원은 각 농가를 나타내는데, 파란색은 청정상태, 붉은 색은 감염상태, 흰색은 살처분상태를 의미한다. 시스템 구성을 위하여 각 농가는 최초 5%의 발병확률을 가지며, 역학보고서에서 감염이 확인된 농가는 해당일자에 발병률이 20%로 설정되면서, 감염 상태로 전환된다. 이후발병률이 증가하여 50%에 이르면 살처분되며, 살처분된 농가는 주변 농가에 영향을 미치지 않는 상황을 일단위로 모의하였다.

지역 내 농가간 전염요인은 크게 계분처리, 동물약품, 가축사료, 농가방문, 입/출하차량, 야생조류로 한정할 수 있다. 선형회귀 분석을 이용하여 이들 6가지 요인에 대한 개별 가중치를 산정하였다. 각 인자 간의 가중치는 계분처리 0.083, 동물약품 0.083, 가축사료 0.215, 입/출하차량 0.782, 농가방문 0.914, 야생조류 -0.079로 산정되어 농가방문, 차량접촉에 의해 매우 큰 영향을 미치는 것으로 확인되었으며, 이 때 잔차는 0.081로 분석되었다(농림수산식품부 2012).

인자간 가중치를 고려하여 각 전염요인의 발생빈도를 결합한 농가간 전염 영향도 행렬을 Figure 49와 같이 도시하였다. 주어진 정방행렬은 시스템을 구성하는 요소들 간의 연결상태를 정의할 수 있다. 주어진 영향도 행렬은 대상지역의 39개 농가 간의 전염 영향을 나타낸 값으로 비대칭행렬로 나타남으로써, 방향성을 고려하여 연결 속성을 정의하였다.

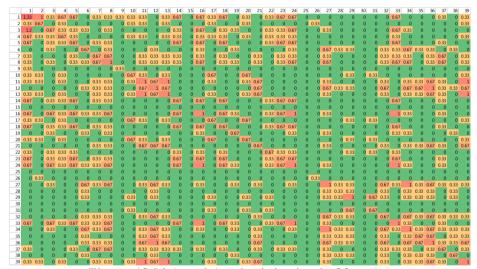


Figure 49 Link weights for infection in 39 nodes

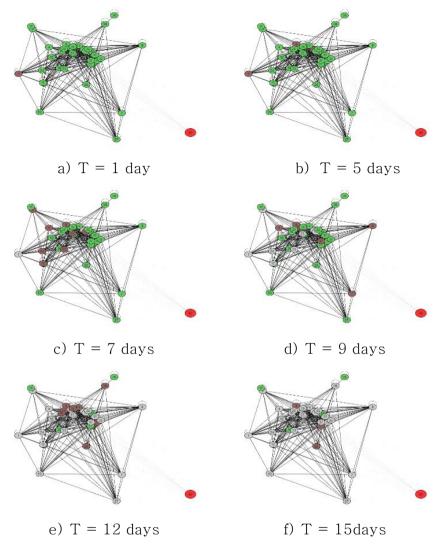


Figure 50 Simulated network about infection status of stockbreeding

모의한 결과는 시간별로 변화하는 모습을 Figure 50과 같이 정리하였다. 초기에는 1번 농가에서만 감염이 되어, 5일이 지날 동안 감염농가가 늘지 않지만, 일주일이 지나면서, 감염농가가 늘고, 늘어난 감염농가의 2차 확산 때문에 급속히 감염농가 수가 늘어난다. 첫 농가 발병 후 약 2주가 지난 15일의 시뮬레이션 결과에서는 대부분 살처분되어 안정세를 보이지만, 많은 농가가 살처분 처리를 피할 수 없었다.

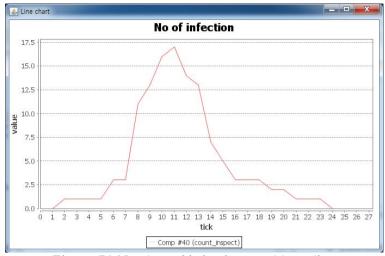


Figure 51 Number of infective stockbreeding

시뮬레이션을 통하여 농가의 감염 수 변화를 살펴보면, Figure 51과 같이 11일이 지난 시점에서 가장 많은 17개의 농가가 감염된 것으로 모의되었으며, 이후 살처분에 따른 전염병 감염 농가 수가줄어드는 모습을 확인할 수 있다. 이와 같이 특정 시점에 감염 개체수가 급격하게 늘어나는 결과는 잠복기를 갖는 병원균 전염의 전형적인 예로 이해할 수 있다. 또한 살처분과 같이 감염 개체가소멸되는 규칙에 의하여 감염 개체수가 급격히 줄어드는 현상도동시에 관찰할 수 있다.

본 절에서는 네트워크로 연결된 농가들 사이의 감염확산을 모의한 문제를 모의하였다. 공기확산만을 고려한 연구에서는 지형적인접성을 기준으로 분석하게 되지만, HPAI 역학보고서에 따르면인접한 농가에 따라 차례대로 감염되는 패턴을 보이지 않고 있다.이는 현상을 지배하는 관계가 공간적 인접성뿐만 아니라, 차량 이동, 사람의 이동과 같은 다른 형태의 연결요소로 인한 인접성이 영향을미치기 때문이다. 병원균 감염농가 확산 모의를 통하여, 본연구에서 개발한 ASAP가 네트워크로 연결된 시스템 문제에서도충분히 적용가능성이 있다고 판단된다.

# 4.6 이질소통을 하는 물리구조

## 4.6.1 문제 선정

물리구조를 갖는 문제는 앞서 유한요소법을 이용하여 풀이가 가능한 편미분 방정식이었다. 동질소통을 하는 경우에는 연결된 요소끼리 전달 메커니즘이 동일했지만, 이질소통 구조를 갖는 경우에는 요소끼리 전달율이 달라지게 된다. 이러한 조건을 갖는 문제로 구조 해석에 관한 문제를 선정하였다. 구조 해석에 있어서 기본 구조물인 트러스를 대상으로 평형방정식을 수립하였으며, 각요소는 연결된 부재와 연결 구조에 따라 연결 속성값을 달리적용하였다.

### 4.6.2 트러스 구조의 해석

농업용 철근콘크리트 구조물의 설계에 있어서 적절한 위치에 철근을 배근하고, 콘크리트 단면의 형상을 최소화하기 위해서는 외부하중에 대한 내부 응력 흐름을 분석해야 한다(윤성수 and 이정재 2002). 일반적으로 스트럿-타이 모델이 가장 널리이용되나, 해석과정에서 연구자의 주관적 견해에 따라 결과가상이할 수 있다는 한계를 갖는다(김윤순 et al. 2004). 이를 개선하기 위하여 연속체 형태의 구조물을 이산체인 트러스 형태로치환하여 해석하는 연구가 이루어졌으며(이성용 et al. 2009), 이러한 환경에서 구조문제는 비교적 단순하고, 비슷한 형태의풀이과정을 반복적으로 거치게 된다. 그러나 기존의 구조 해석방법은 문제를 해석할 때마다 강성행렬을 새롭게 생성하여풀이함으로써(김시환 et al. 2011), 계산의 효율성이 떨어지고, 해석과정이 복잡한 한계를 가진다.

이한귀 등은 이러한 원인에 대하여 절차지향적 프로그래밍 방식의 한계와 독립성이 보장되지 않는 객체지향적 프로그래밍 방식에 기인한다고 주장하였다(이한귀 et al. 2012). 독립적으로 계산 가능한 구조요소와 반복계산방법을 개발하여, 구조요소를 연결하는 작업을 통하여 전체 강성행렬을 만들지 않고 보 구조물을 해석하였다. 특히 요소 독립성이 보장되도록 객체지향적으로 구현함으로써, 구조해석 도중에 구조물의 요소가 결합되더라도 효과적으로 문제를 해결할 수 있음을 검증하였다. 이는 구조물에 대한 기존 해석 결과가 존재하는 경우, 유사한 구조물에 대하여 효과적인 해석이 가능함을 의미한다.

본 연구에서는 트러스 구조 해석을 위한 요소 독립성이 유지되는 객체기반 구조해석모델을 개발하였다. 모델의 구현을 위하여 독립성이 유지되는 시뮬레이션 환경으로 ASAP를 이용하였으며, ASAP 시뮬레이션 환경에서 연결방법 및 컴포넌트 구현 방식을 개선하고, 구조해석 컴포넌트의 연결을 통하여 트러스 구조물을 해석하였다. 해석된 결과는 기존 프로그램을 이용한 결과와 비교를 통하여 검증하였다.

### 1) 트러스 구조의 평형방정식

트러스는 힌지로 연결된 봉 부재로, 부재 방향으로 1차원 내력만 발생하며, 양쪽 끝단에서 모멘트에 저항하지 않고, 변위만 발생한다. 매트릭스 구조 해석 방법에서는 2차원 트러스를 해석하기 위하여 트러스를 하나의 요소로 설정하여 4x4 강성행렬을 만들어 평형방정식을 풀이한다. 본 연구에서는 독립적으로 해석가능한 컴포넌트를 구성하기 위하여 트러스를 하나의 요소로 설정하지 않고, 노드를 중심으로 평형방정식을 정식화하였다.

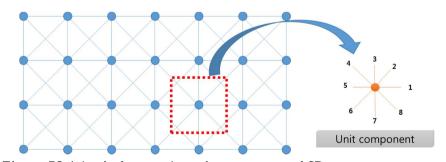


Figure 52 A basic form and a unit component of 2D truss structure

기본적인 구조물의 형태가 Figure 52와 같이 구성된다고 가정(Gürdal and Brian 2000)하면, 각 노드는 여러 개의 트러스와 연결된다. 따라서 각 노드에서 평형방정식은 식 (53), (54)와 같다.

$$\sum_{k=1}^{n} F_{xk} + f_x = 0 (53)$$

$$\sum_{k=1}^{n} F_{yk} + f_{y} = 0 {(54)}$$

여기서, n은 노드에 연결된 트러스의 개수,  $F_{xk}$ ,  $F_{yk}$ 는 k번째 트러스의 x, y 방향 내력,  $f_x$ ,  $f_y$ 는 각 노드에 작용하는 외력을 의미한다. 노드에서 각 트러스 방향을 나타내는 방향 벡터  $(\cos\theta_k, \sin\theta_k)$ 를 고려하면, 각 트러스에 작용하는 x, y 방향 내력은 식 (55), (56)과 같이 트러스에 작용하는 내력인 인장력혹은 압축력으로 정리할 수 있다.

$$F_{xk} = F_k \cos \theta_k \tag{55}$$

$$F_{yk} = F_k \sin \theta_k \tag{56}$$

각 트러스에 작용하는 내력은 탄성방정식에 따라 식 (57)과 같이 정리할 수 있다.

$$F_k = EA_k \varepsilon_k \tag{57}$$

여기서, E 는 탄성계수이고,  $A_k$  는 k번째 트러스의 단면적을 의미한다. 또한  $\varepsilon_k$ 은 변형률을 의미하는데, 트러스는 길이방향으로 변형되고, 미소 변형의 원리를 고려할 때, 식 (58)과 같이 정리할 수 있다.

$$\varepsilon_k = \frac{\Delta l_k - l_k}{l_k} \tag{58}$$

여기서,  $\Delta l_k$ 는 트러스의 변형량으로 노드의 변위로 구할 수 있으며,  $l_k$ 는 트러스의 길이로 미소변형의 원리에 따라 초기 노드 위치에

따라 결정된다. 이를 정리하면, 식 (59)와 같이 변형률을 노드의 변위로 기술할 수 있다.

$$\varepsilon_{k} = \frac{(u_{k} - u_{n})\cos\theta_{k} + (v_{k} - v_{n})\sin\theta_{k}}{\sqrt{(x_{k} - x_{n})^{2} + (y_{k} - y_{n})^{2}}}$$
(59)

여기서,  $u_n$ ,  $v_n$ 은 기준 노드에서의 x, y방향의 변위를 의미하며,  $u_k$ ,  $u_k$ 는 기준 노드에서 k번째 트러스로 연결된 반대쪽 노드에서의 x, y방향 변위를 의미한다.  $x_n$ ,  $y_n$ 은 기준 노드의 위치 좌표를 의미하며,  $x_k$ ,  $y_k$ 는 각각 k번째 트러스로 연결된 반대쪽 노드의 좌표를 의미하다.

#### 2) 트러스 구조 컴포넌트의 구현

앞서 트러스 구조해석을 위하여 절점을 분석 단위로 설정하고, 절점이 8개의 트러스를 갖는 구조로 지배방정식을 정식화하였다. 이에 따라 트러스절점 컴포넌트는 Figure 53과 같이 구현하였다. 만약 트러스절점이 경계조건으로 고정단인 경우에는 변위가 발생할수 없으므로, 계산과정을 생략하고, 처음 조건을 그대로 유지하게된다. 따라서 3번째 라인처럼 절점컴포넌트의 속성 중 고정단으로 표시해둔 의미로 "fixed" 속성이 존재하는 경우에 계산과정을 생략하고, process() 메소드에서 빠져 나온다.

트러스노드 컴포넌트가 일반 노드인 경우에는 주변 트러스 8개의 합력과 해당 노드의 변위를 산정해야 한다. 앞서 식 (53), (54)과 같이 2개의 방정식을 만족하는 변위 값 u, v를 계산해야 하므로 미지수가 2개인 1차 연립방정식이 된다. 이는 2x2 행렬과 2x1 행렬의 곱으로 계산되므로, 2x2 행렬을 저장하기 위한 배열 uv 변수와 2x1 행렬을 저장하기 위한 sumB 변수를 코드 4째 줄부터 10째 줄에서 정의하고, 초기화하였다.

코드에서 11번째 줄부터는 인접 컴포넌트의 값을 이용하여 행렬 값을 채우는 과정을 서술하였다. 이렇게 채워진 값은 postProcess()에서 역행렬을 구함으로써, 해당 노드에서의 변위 u, v를 산정할 수 있다.

```
01:
     @Override
02:
     public void process(List<TreeNode> neighborhoods) {
03:
         if (getString("fixed") != null) { return; }
04:
         sumB[0] = getDouble("fx");
05:
         sumB[1] = getDouble("fy");
06:
         for (int i = 0; i < 2; i++) {
07:
             for (int j = 0; j < 2; j++) {
08:
                 uv[i][j] = 0;
09:
             }
10:
11:
         for (TreeNode n : neighborhoods) {
12:
             IASAPComponent nm = n.getModel();
13:
             double x2 = nm.getDouble("x");
14:
             double y2 = nm.getDouble("y");
15:
             double x1 = getDouble("x");
16:
             double y1 = getDouble("y");
17:
             double A = getDouble("A", nm.getId());
18:
             double E = 10000.;
19:
             double u = nm.getDouble("u");
20:
             double v = nm.getDouble("v");
21:
             double L = Math.sqrt(Math.pow(x2-x1, 2) + Math.pow(y2-y1, 2));
22:
             double cos = (x2 - x1) / L;
23:
             double sin = (y2 - y1) / L;
24:
             sumB[0] += (E*A*cos*u/L + E*A*sin*v/L) * cos;
25:
             sumB[1] += (E*A*cos*u/L + E*A*sin*v/L) * sin;
26:
             uv[0][0] += E*A*cos/L*cos;
27:
             uv[0][1] += E*A*sin/L*cos;
28:
             uv[1][0] += E*A*cos/L*sin;
29:
             uv[1][1] += E*A*sin/L*sin;
30:
         }
31:
     }
```

Figure 53 A main program code for truss node component

#### 3) 10-트러스 구조의 해석

구현한 트러스 노드 컴포넌트를 이용하여 트러스를 해석할 수 있는지 검증하기 위하여, Gürdal and Brian (2000)이 제시한 문제를 Figure 54와 같이 적용하였다. 컴포넌트의 속성으로 노드의 좌표, 트러스 단면적, 하중, 경계조건, 탄성계수를 설정하였다. 탄성계수는 10,000 으로 가정하여 계산한 결과, Figure 54의 오른쪽 그림과 같이 처짐이 발생하였고, 2번 노드에서 변위가 u=-7.366855, v=-18.021140, 3번 노드에서 변위가 u=-9.522367, v=-39.395724로 계산되어, Gürdal and Brian (2000)이 제시한 결과와 비교하여 유효숫자 내에서 정확히 일치하였다. 결과를 얻는 데까지 필요한 반복횟수는 Mathematica®를 이용하여 703번 반복한 반면, 본 연구에서는 동일한 결과를 구하기 위해 오차범위를  $10^{-6}$ 으로 설정하여, 357번 반복계산을 통하여 결과를 산출하였다.

결과에서 특징적으로 반복계산을 수행하는 동안 처짐이 화면 상에 그림으로 반영되어 직관적으로 결과를 이해하는 데 도움을 준다.

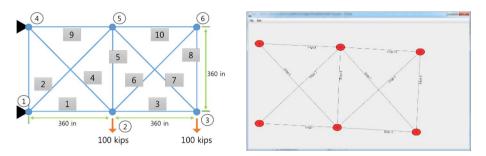


Figure 54 Ten bar problem and solution (This study)

#### 4) 5-트러스 구조의 해석

개발한 트러스 구조 컴포넌트의 적용 예를 살펴보기 위하여, 모델개발 과정에 가정한 8방향 형태의 트러스와는 다른 형태인 삼각형 트러스 구조를 풀이하였다. 결과의 검증을 위하여 프로그램으로 해석한 결과가 제공되는 문제를 선정하였으며, 문제의 형상은 Figure 55와 같이 도시할 수 있다. 트러스와 노드의 속성은 Table 7와 같이 정리할 수 있으며, 해석 결과는 웹을 통해 Matlab® 코드를 제공하고 있다(Bhatti 2005).

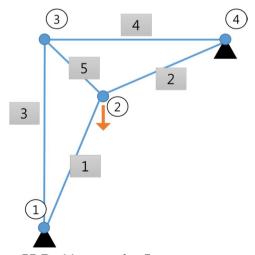


Figure 55 Problem set for 5-truss structure

Table 7 Properties of nodes and edges for 5-truss problem

Node ID	x (cm)	y (cm)	Element ID	E (GPa)	A (cm <sup>1</sup> )
1	0	0	1	200	40
2	150	350	2	200	40
3	0	500	3	200	30
4	500	500	4	200	30
			5	70	20

제공된 코드를 이용하여 주어진 문제의 변위를 계산하였다. 문제에서 2번 절점에 작용하는 하중을 150kN으로 정의하였을 때, 고정단은 입력조건에 따라 변위가 없고, 고정단을 제외한 2번 절점과3번 절점에서의 변위는 각각 (0.5390, -0.9531), (0.2647, -0.2647)로 계산되었다.

본 연구에서 개선한 ASAP 환경에서 문제를 해결하기 위하여, Figure 56과 같이 문제의 조건을 프로그램 코드로 작성하였다. 메인 시뮬레이션 환경인 ASAPEnvironment 인스턴스에 구동하기 위한 컴포넌트 객체를 생성하여 알려주면. ASAPEnvironment가 각 컴포넌트 객체를 주어진 정보에 따라 연결하여 시뮬레이션을 수행한다. 코드에서 살펴보면, 2-6라인에서 앞서 설계한 트러스 구조 컴포넌트를 생성하고, 7-21라인에서 트러스 노드의 속성인 좌표, 고정단 여부, 그래프 표시 여부 등을 정의한다. 22라인에서 ASAPEnvironment 인스턴스에 컴포넌트 정보를 전달함으로써, 시스템을 구성하는 요소에 대한 정의는 마무리된다. 이후 코드는 요소들 간의 관계를 정의하기 위한 코드로서, 24-26라인은 인접행렬을 배열로 정의하였다. 27-45라인까지는 연결 속성, 즉 트러스 속성을 의미하는 단면적, 탄성계수를 정의하고 있다. 제시한 코드와 같이 연결속성을 모두 정의하고 나면, 요소에 대한 정의와 마찬가지로 46라인에서 ASAPEnvironment 인스턴스에 요소의 연결정보를 전달한다.

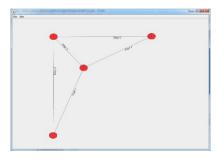
```
01:
     public void problem5Truss(ASAPEnvironment app) {
02:
         int nodeCount = 4;
03:
         IASAPComponent[] nodes = new IASAPComponent[nodeCount];
04:
         for (int i = 0; i < nodeCount; i++) {</pre>
05:
             nodes[i] = new GASSTrussComponent(i+1);
06:
07:
         nodes[0].setDouble("x", 0);
08:
         nodes[0].setDouble("y", 0);
09:
         nodes[0].setString("fixed", "fixed");
10:
         nodes[1].setDouble("x", 150);
11:
         nodes[1].setDouble("y", 350);
12:
         nodes[1].setDouble("fy", -150);
13:
         nodes[1].setString("plot_item", "v");
         (\ldots)
22:
         app.setNode(nodes);
23:
24:
         int[][] adjacencyMat = new int[][]{
25:
             (-1,0,2,-1), (-1,-1,4,1), (-1,-1,-1,3), (-1,-1,-1,-1)
26:
         };
27:
         for (int i = 0; i <adjacencyMat.length; i++) {</pre>
28:
             for (int j = 0; j <adjacencyMat[i].length; j++) {</pre>
29:
                 if (adjacencyMat[i][j]>=0 &&adjacencyMat[i][j]<2) {</pre>
30:
                     nodes[i].setDouble("A "+nodes[j].getId(), 40);
31:
                     nodes[j].setDouble("A_"+nodes[i].getId(), 40);
32:
                     nodes[i].setDouble("E "+nodes[j].getId(), 200);
33:
                     nodes[j].setDouble("E_"+nodes[i].getId(), 200);
34:
                 } else if (adjacencyMat[i][j]>=2&&adjacencyMat[i][j]<4) {</pre>
                 (...)
39:
                 }
                 (...)
             }
44:
45:
         app.setAdjacencyMatrix(adjacencyMatrix);
46:
47:
```

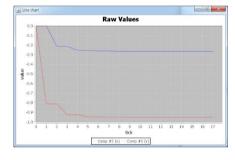
Figure 56 A main program to set up problem

제시한 코드를 이용하여 시뮬레이션을 수행하면, Figure 57과 같은 결과가 화면에 표시된다. 코드에서 입력한 컴포넌트의 위치 및 인접행렬을 이용하여 Figure 57(a)와 같이 네트워크 형태로 문제를 도시해 준다. 그리고 주어진 조건에 따라 시뮬레이션한 결과 Figure 57 (b)와 같이 결과를 실시간으로 그래프로 표시해 준다. 그래프는 앞서 Figure 56에서 작성한 코드에서 컴포넌트 속성으로 "plot\_item" 이라는 변수에 원하는 변수 이름(여기서는 y축 변위량인 "v"를 입력함)을 매개변수로 넘겨주면, 자동으로 값의 변화를 도시한다. 앞서 Matlab® 코드의 해석 결과 (-0.9531 mm, -0.2647 mm)와 같은 결과로 점근하고 있음을 Figure 57 (b)에서

확인할 수 있다. ASAPEnvironment에서 모의된 모든 값은 sqlite라는 파일 데이터베이스로 저장되는데, 계산결과 기존 연구결과와 유효숫자 내에서 동일한 결과를 확인하였다.

고정단을 제외한 두 절점인 2번 절점과 3번 절점의 y방향 변위를 도시한 Figure 57 (b)는 반복계산횟수가 증가함에 따라 특정 값에 수렴하는 형태를 보여주었다. 이는 두 절점이 주변의 응력에 따라 자기의 위치를 주도적으로 변화시키면서 적응하는 행태로 이해할 수 있다. 그래프를 좀 더 자세히 살펴보면, 2번 절점에 작용하는 힘에 의해서 2번 절점이 먼저 움직이게 되고, 다음 반복계산 시 2번 절점의 변위에 따라 3번 절점도 평행을 이루기 위해서 움직인다. 반복계산이 진행됨에 따라, 3번 절점의 변위로 인해 2번 절점이 다시 평행을 이루기 위하여 위치를 이동하며, 이와 같은 반복 수행을 통하여 모든 절점이 평형상태에 도달하게 된다.





(a) Network Viewer

(b) Chart Viewer for results

Figure 57 Results of 5-truss problem using ASAP

### 5) 11-트러스 구조의 해석

대표적인 구조해석 프로그램 중 하나인 ANSYS®를 이용한 해석 결과(University of Alberta 2001)와 비교하기 위하여 Figure 58과 같이 11-트러스 구조(Chandrupatla and Belegunda 2002)를 해석하였다. 문제에서 트러스는 모두 동일한 소재로서, 탄성계수는 200 GPa, 단면적은 32.5  $cm^2$  으로 설정하였다. 특징적으로 11-트러스 구조에서는 7번 절점이 롤러 고정단으로 y축으로는 변위가 없고, x축으로는 변위를 가질 수 있다는 점에서 이전 문제와는 구별된다.

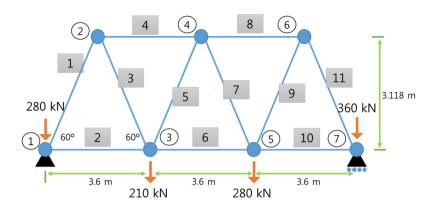


Figure 58 Example for 11-truss structure

상용 구조해석 프로그램인 ANSYS® 7.0를 이용하여 해석한 결과와 본 연구에서 개발한 트러스 구조 컴포넌트를 이용한 시뮬레이션 해석 결과를 Table 8에 정리하였다. 분석된 변위량은 모든 절점에서 유효숫자 내 동일한 값을 나타냈다.

Table 8 Displacements comparison of ANSYS and this study

Node	ANSYS®		GASSTrussComponent	
	Ux	Uy	U	V
1	0.0000	0.0000	0.00000	0.00000
2	3.0836	-3.5033	3.08364	-3.50328
3	0.7460	-6.5759	0.74604	-6.57587
4	1.5916	-7.2363	1.59155	-7.23630
5	2.3127	-6.9923	2.31273	-6.99228
6	-0.0497	-3.7330	-0.04973	-3.73301
7	3.1334	0.0000	3.13337	0.00000

### 6) 기존 해석 프로그램과의 비교

이상에서 살펴본 바와 같이 ASAP를 이용하여 트러스 구조를 정확하게 해석할 수 있음을 확인하였다. 기존 구조해석과 ASAP를 이용한 구조해석의 차이를 비교하기 위하여, 기존 구조해석 프로그램의 대표적인 방법인 유한요소법(이정재 1991)과 ASAP 트러스 모델의 순서흐름도를 Figure 59와 같이 도시하였다.

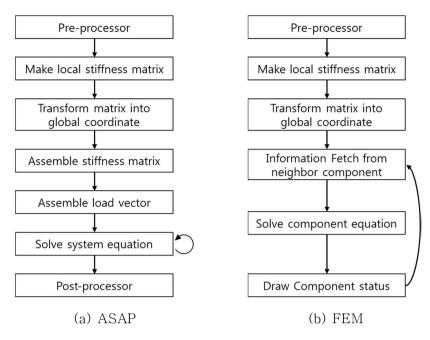


Figure 59 Flow chart of FEM and truss simulation with ASAP

기존 구조해석 프로그램은 전체 강성행렬을 만든 후, 행렬을 계산하고 그 결과를 후처리 과정에서 절점과 트러스에 배분하는 구조를 갖는다. 반면, ASAP를 이용한 트러스 구조해석 모형은 각노드에서 지배방정식을 만든 후, 주변 절점에서 정보를 취득하여 대상 노드의 지배방정식을 계산하고, 그 결과를 화면에 반영한 후, 다시 주변에서 변경된 정보를 갱신하여 지배방정식을 계산하는 과정을 반복한다. 이는 기존 프로그램에서 강성행렬을 반복법으로 푼다면 동일한 해석과정으로 볼 수도 있고, 행렬 계산을 위한다양한 알고리즘을 적용할 수 있는 기존 방식이 계산 속도 및효율은 더 좋다고 평가할 수 있다.

그러나 전체 강성행렬을 만들어서 풀게 되면, 전체 강성행렬을 만드는 조합과정이 매우 번거롭고, 전체 강성행렬을 해석할 때까지결과를 알 수 없다는 단점이 있다. 반면 ASAP 환경에서는 독립적으로 컴포넌트가 해석될 수 있으며, 연구자는 단순히 절점의속성과 트러스의 속성을 입력함으로써 손쉽게 시뮬레이션 수행이가능하다. 시뮬레이션이 수행되는 동안 컴포넌트 내에서 속성값이변화하는 과정을 시각화함으로써 현상을 이해하는 데 도움을 준다.

또한 트러스가 정적 평형상태를 이루지 못하고 강체 운동(rigid body motion)을 보이는 경우, 기존 프로그램은 전체 강성행렬을 생성하여 해석하므로 행렬을 계산할 수 없다는 정보만을 사용자에게 제공한다. 반면 본 연구의 결과물은 반복해법을 적용하여 실시간으로 계산결과를 시각화하기 때문에 구조물이 형상을 유지하지 못하고, 값이 발산하는 시각 정보를 통하여 형상의 구조적 문제가 있음을 연구자에게 쉽게 전달한다.

향후 트러스 구조 컴포넌트는 구조해석의 기본적인 형태로서, 확장을 통하여 트러스와 빔 구조가 결합된 문제 혹은 판(plate) 구조 문제 등의 기초 모델로 활용 가능할 것으로 판단된다. 또한 변형을 실시간으로 시각화함으로써 설계자가 구조물 설계과정에서 즉각적으로 피드백을 얻을 수 있어 대화형 설계가 가능할 것으로 사료된다.

# 4.7 복제기능을 통해 성장하는 시스템

### 4.7.1 문제 선정

복제 기능은 시스템을 구성하는 요소를 생성함으로써, 시스템의 외연을 확장을 가능케 한다. 앞서 상술한 예제는 사용자가 정해놓은 시스템 내부의 메커니즘을 모의하였으나, 본 절에서는 새로운 컴포넌트를 생성하여 시스템 자체가 성장하는 문제를 풀이하고자하였다. 주변에서 쉽게 찾아볼 수 있는 생장하는 시스템으로 생물체가 대표적이며, 그 중에서도 비교적 시뮬레이션 모델에 관한연구가 활발한 초목의 생장모델을 재현 예제로 선정하였다. 또한기능적으로 시스템을 구성하는 전처리기로 활용 가능하므로, 동일한컴포넌트를 다수 생성해야하는 등가트러스 모형을 두번째 예제로 선정하였다.

초목의 생장모형은 내부 속성값의 변화보다는 시스템 외연이 확장되어가는 도메인의 형상이 주요한 결과물이므로, 연결상태를 부각하여 결과를 도시하였으며, 속성값의 시계열 변화를 나타내는 그래프는 생략하였다. 등가트러스 모형은 구조분야에서 검토한 셀룰러 오토마타를 이용한 트러스 컴포넌트를 이용하므로, 컴포넌트 개발 및 계산과정에 대한 설명은 간소화하고, 형상을 생성하기 위한 전처리 기능 중심으로 서술하였다.

## 4.7.2 초목의 성장모의

초목의 성장모형은 앞서 cloning 기능을 검토하는 과정에서 언급한 바가 있다. 1960년대 생물학 분야에서 Lindenmayer (1968)가 초목의 형태를 설명하기 위하여 L-systems을 소개한 이후, 이를 기반으로 다양한 연구에서 활용되고 있다. 앞서 L-systems를 구현하는 과정에서 나뭇가지의 길이, 각도를 고정하여 시뮬레이션을 수행하였으나, 본 절에서는 보다 사실적인 생장을 모의하기 위하여 확률변수를 이용하여 개발하였다.

개발한 컴포넌트의 주요 코드는 Figure 60과 같다. 해당 코드는 크게 3가지 메소드로 구성되는데, 상단의 process() 메소드에서는 가지의 각도, 가지의 길이 등 컴포넌트의 거동 속성을 결정하는 기능을 서술하였다. 코드 가운데 위치한 preProcess()는 나뭇가지의 성장을 멈추기 위한 조건을 기술하였으며, 코드 하단의 makeBranch() 메소드는 주어진 각도와 길이에 따라 컴포넌트를 복제하여 새로운 나뭇가지 컴포넌트를 생성하는 기능을 수행한다.

```
@Override
01:
02:
     public void process(List<TreeNode> neighborhoods) {
03:
         if (!isComplete_) {
04:
            double angle = Math.random()*15+30;
05:
            double branchLength = get("length") * (Math.random()*.2+.9);
06:
            if (neighborhoods.size() == 0) {
07:
                makeBranch(5-Math.random()*10, branchLength * .8);
08:
            } else {
09:
                makeBranch(5-Math.random()*10, branchLength * .8);
10:
                makeBranch(angle, branchLength * (0.6+Math.random()*.2));
11:
            }
12:
            isComplete = true;
13:
14:
15:
     @Override
16:
     public void preProcess() {
         if (get("length") < 10) {</pre>
17:
18:
            isComplete = true;
19:
         }
20:
21:
     public void makeBranch(double angle, double branchLength) {
22:
         IASAPComponent branch = new
23:
              LindermayerSystemRandModel(universe .getMaxId());
         double dirx = get("dirx");
24:
25:
         double diry = get("diry");
26:
         double dirxy = Math.sqrt(Math.pow(dirx, 2) + Math.pow(diry, 2));
27:
         dirx /= dirxy;
28:
         diry /= dirxy;
29:
         double sin = Math.sin(get("prefSide") * Math.PI / 180 * angle);
30:
         double cos = Math.cos(get("prefSide") * Math.PI / 180 * angle);
31:
         branch.set("x", get("x")
32:
                           + (dirx * cos + diry * sin) * branchLength);
33:
         branch.set("y", get("y")
34:
                          + (-dirx * sin + diry * cos) * branchLength);
         branch.set("prefSide", get("prefSide") * -1);
35:
36:
         branch.set("size", 0.05);
37:
         branch.set("length", branchLength);
38:
         branch.set("dirx", dirx * cos + diry * sin);
39:
         branch.set("diry", -dirx * sin + diry * cos);
40:
         universe_.addChild(this, branch);
41:
     }
```

Figure 60 A main program code for plant growth

먼저 나뭇가지 컴포넌트의 속성을 확률을 이용하여 결정하는 상단의 코드를 살펴보면, 각도는 30-45도 사이에서 균등확률 분포를 통해 값을 결정하며, 나뭇가지의 길이는 기준값으로 0.9-1.1배 사이의 값을 갖는다. 이 기준값을 이용하여, 나뭇가지가 자라나는 방향으로는 -5도에서 5도 사이로 나뭇가지가 80% 길이만큼 성장하며, 곁가지는 60-80% 사이의 길이로 좌우 교대로 앞서 결정된 각도 방향으로 뻗어나가도록 구현하였다.

성장하는 문제에서는 성장이 멈추는 종료조건이 중요하다. 구현한 코드에서 한번 뻗은 나뭇가지 컴포넌트는 복제기능을 수행하고 난 후, 성장을 멈추도록 구현하였다. 그리고 뻗어나간 나뭇가지의 길이가 10 이하가 되면, 복제 기능을 수행하지 않기전에 작동을 멈추도록 구현하였다.

코드 하단에서는 새로운 컴포넌트를 생성하고, 나뭇가지 컴포넌트에 필요한 속성으로 방향, 길이, 가지가 뻗어나갈 선호방향 등을 정의하였다. 나뭇가지의 각도는 degree로 표현하였으며, 35번째 줄에서 선호하는 방향에 따라 좌우방향이 전환되도록 코드를 작성하였다. 모든 속성을 복제된 컴포넌트에 입력한 후, 40번째 줄에 작성한 코드처럼 ASAPEnvironment에 복제된 컴포넌트를 전달함으로써 다른 컴포넌트에서 새롭게 생성된 컴포넌트를 인지할 수 있도록 위임하였다.

작성한 컴포넌트를 이용하여 시뮬레이션을 구동하면, 한점에서 시작한 작은 점에서 연결선이 늘어나면서, 초목이 완성되어가는 과정을 확인할 수 있다. 모의하는 과정에서 난수를 이용하여 나뭇가지의 길이와 각도가 결정되므로, 결과물은 Figure 61의 (a), (b)와 같이 항상 다른 형상을 보여준다. 본 절에서는 간단한 규칙을 이용하여 형상만을 모의하였지만, 초목의 생장에 관한 메커니즘을 고려하여 보다 현실적인 모형으로 확장 가능할 것으로 기대한다.

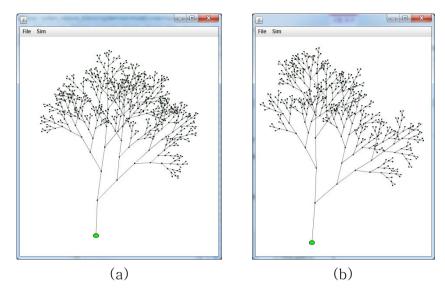


Figure 61 Simulation for plant growth based on L-systems

### 4.7.3 전처리 기능을 활용한 등가트러스 해석

등가트러스 모형은 연속체 구조물의 확률 해석의 계산량을 줄이기 위하여 연속체 구조물을 동일한 거동을 보이는 등가의 트러스로 치환하여 구조물을 해석한 모델이다(최원 2012). 구조물 해석에 이용되는 유한요소법(FEM, Finite element method)이 메쉬를 생성하여 해석하는 것과 마찬가지로 등가의 트러스 모형도 연속체 구조물을 다수의 트러스로 표현해야 한다. 트러스의 개수가 많을수록 정해에 가까워지므로(이성용 et al. 2009), 컴퓨터 계산성능과 허용오차율을 고려하여 최대한 많은 양의 트러스가 요구된다. 이에 다수의 트러스 컴포넌트를 효과적으로 생성하고 배치하는 방안으로 복제 기능을 활용하였다.

등가 트러스를 해석하기 위한 컴포넌트는 앞서 개발한 셀룰러 오토마타를 이용한 트러스 컴포넌트를 이용하였으며, 연속체와 등가의 거동을 모의하기 위하여 트러스의 단면적을 식 (60)과 같이 적용하였다(Tatting and Gürdal 2000).

$$A_o = \frac{3}{4}lt$$
  $A_d = \frac{3\sqrt{2}}{8}lt$  (60)

깊은 보를 예제로 선정하여, cloning 기능을 통하여 형상을 구성하였다. 화면 상에 하나의 컴포넌트를 위치시키고, 오른쪽으로 50세대, 위쪽으로 25세대 복제가 되게끔 지정하였다. 그리고 시뮬레이션을 실행하여, 복제규칙에 따라 예제로 선정한 깊은 보형상을 완성하였다. 복제가 끝난 후, 주어진 문제에 외부하중조건을 입력하고, 다시 시뮬레이션을 실시하여, 구조해석을 수행하였다. 각 컴포넌트의 색은 내부 에너지량으로 표현하도록하였으며, 내부 응력은 식 (61)과 같이 정의하였다.

$$U = \left[A_o \varepsilon_h^2 + A_o \varepsilon_v^2 + \sqrt{2} A_d (\varepsilon_{d1}^2 + \varepsilon_{d2}^2)\right] \frac{El}{2}$$
 (61)

외부하중을 가하여 시뮬레이션을 실시한 결과, Figure 62와 같이 결과를 내부 응력 분포를 확인할 수 있었다. 해석 결과는 Tatting and Gürdal (2000)가 제시한 해석결과와 동일한 패턴의 분포를 확인하였다.

GASS는 물리구조에 기반하여 개발되었으나, 다수의 요소를 화면 상에 위치시키기 어려운 기술적 한계로, 집체 모형(lumped model)으로 주로 모델링이 이루어졌다. 그러나 ASAP에서는 복제기능을 제공함으로써, 다수의 컴포넌트를 정해진 규칙에 맞추어생성할 수 있게 되었다. 따라서 간단한 그리드 형태뿐만 아니라, 앞서 살펴본 수목의 성장과 같이 규칙을 갖는 복잡한 구조 역시쉽게 모델링이 가능할 것으로 판단된다.

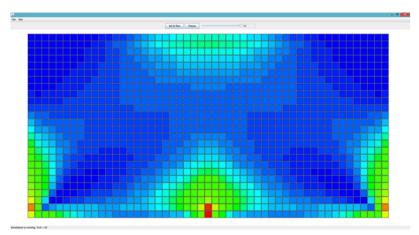


Figure 62 Simulated internal energy for deep beam to solve by equivalent truss component

### 4.8 소결

본 장에서는 개발된 ASAP를 이용하여 다양한 예제를 구현함으로써 활용성을 검증하고자 하였다. 시스템 시뮬레이션 모형을 구현하기 위해서는 컴포넌트를 개발하는 과정과 컴포넌트를 이용하는 과정이 필요하다. 모델 컴포넌트는 IASAPComponent를 상속하여 구현하여야 하며, 구현의 편의성을 위하여 DefaultModel 클래스를 제공하고 있다. 추상클래스인 DefaultModel를 상속하여, 개발하고자 하는 모형의 메커니즘을 java 코드로 작성해야 한다. 작성된 컴포넌트는 연결 정보와 컴포넌트가 갖는 속성값을 입력한 후, 시뮬레이션 환경인 ASAPEnvironment 클래스를 구동하여 결과를 얻을 수 있다.

개선된 ASAP의 기능을 확인하기 위하여 위상구조와 물리구조, 동질소통과 이질소통, 그리고 복제기능에 관한 예제를 선정하여 시뮬레이션을 실시하였다. 각 예제는 지역시스템공학 과 관련한 다양한 분야에서 선정하였으며, 선정된 문제는 지배방정식의 정식화 과정을 거쳐 컴포넌트를 개발한 코드를 제시하여, 컴포넌트 개발과정을 설명하고자 하였다. 대부분의 예제에서 작성한 주요 코드가 50줄을 넘지 않았으며, 개발된 컴포넌트가 존재하는 경우에는 코드를 재활용할 수 있으므로 개발과정이 어렵지 않다고 판단하였다.

개발한 컴포넌트를 이용하여 시스템 시뮬레이션을 수행하면, 문제 도메인을 나타내는 그림과 특정 컴포넌트의 속성을 차트로 표현해주는 2개의 창을 확인할 수 있다. 이들 창을 통해 시뮬레이션이 수행되는 과정에서 실시간으로 결과를 확인할 수 있다. 각 예제에서 분석한 결과는 참조한 연구결과와 동일한 결과를 확인하였다. 그리고 모델을 개선하기 위해서는 전체 시스템을 수정하지 않고, 개발한 컴포넌트의 일부 코드만 개선하면, 시스템 전체가 향상되는 결과를 확인하였다. 초목의 성장모형은 3장에서 제시한 L-systems 컴포넌트에서 확률변수만 추가하여 모의한 결과로 상대적으로 사실적인 초목형상을 모의할 수 있는 모델로 개선되었으며, 복제 기능은 컴포넌트의 생성 및 배열을 손쉽게 처리할 수 있는 전처리 도구로써 활용 가능하였다.

# 제 5 장 결 론

## 5.1 요약 및 결론

본 논문에서는 다양한 시스템 시뮬레이션 도구를 검토하고, 이들 시뮬레이션 도구가 갖는 한계를 지적하고, 지역시스템에 가장적합한 시뮬레이션 도구 중 하나인 GASS를 기반으로 하여, 3가지측면에서 개선한 ASAP를 개발하였다. 개선된 3C 기능인연결(Connection), 소통(Communication), 복제(Cloning) 기능에대하여 예제를 통하여 검토하였으며, 개발된 ASAP를 이용하여다양한 지역시스템 문제에 적용하여, 활용가능성을 평가하였다.

연결(Connection) 기능의 개선을 통하여 물리구조적 문제인 열전도 문제뿐만 아니라, 위상구조를 갖는 Stochastic PERT 문제, 더 나아가 물리구조와 위상구조가 혼합된 제어기가 설치된 열전도 문제를 모의하였다.

소통(Communication) 기능의 개선을 통하여 동질소통구조로 정보를 교환하는 트러스의 부재력 문제와 이질소통구조로 정보를 교환하는 모멘트 분배법을 구현하여 시뮬레이션하였다.

복제(Cloning) 기능의 개선을 통하여 식물의 생장 모형인 Lindenmayer systems를 구현하고, GASS에서는 불가능했던 시스템의 외연의 확장 문제에 대하여 검토하였다.

시뮬레이션 플랫폼의 활용성을 검토하기 위하여 다양한 분야의 지역시스템공학 문제를 선정하고, 각 문제의 거동을 모의하기 위한 컴포넌트를 개발하여 시뮬레이션을 수행하였다. 선정된 문제는 본 연구에서 개선한 3C 기능을 중심으로 분류하여 고찰하였다.

수리/수문분야 문제로는 연결된 저수조의 수위 해석과 삼형제 저수지의 수위 모의를 수행하였다. 해당 분야는 시스템을 구성하는 위치정보보다는 연관관계가 더 중요하며, 모델 간의 소통구조가 단일한 형태의 정보교환이 이루어지는 동질소통구조를 갖는다. 여기서, 실세계에 존재하는 대상체임에도 불구하고, 물리구조가 아닌 위상구조로 다루어지는 문제는 주로 집체 모형(lumped model)으로 설명되는 문제들이다.

연결된 저수조의 수위 해석은 저수조 1개에서 위치에너지와

운동에너지의 변환을 이용하여 지배방정식을 도출하고, 저수조가 2개 연결된 상태에서 일반화된 컴포넌트를 개발하였다. ASAP를 이용하여 시뮬레이션을 실시하여, 미분연립방정식의 해석해와 비교함으로써 모델을 검증하였다. 검증된 컴포넌트를 이용하여, 연결된 저수조 3개의 수위를 모의함으로써 미분방정식과 달리도메인이 커지더라도 해석에 어려움이 없음을 제시함으로써 수학적/공학적 해석기법과 시뮬레이션 기법의 차이를 확인하였다.

삼형제 저수지는 저수조 컴포넌트를 기반으로 유입, 관개, 증발, 침투량, 수위-내용적을 고려한 저수지 컴포넌트로 확장하여 개발하였다. 기상자료 컴포넌트, 유역 컴포넌트, 수혜지역 컴포넌트 등을 추가하여, 서로 연결되어 있는 용계, 무극, 금석 저수지의 수위를 모의하였으며, 용계저수지와 금석저수지에서 실시간으로 관측한 수위데이터와 비교함으로써, 독립적으로 개발된 컴포넌트를 이용한 객체기반 시뮬레이션의 효용을 검토하였다.

지역시스템 공학 분야에서 현상을 설명하기 위해서 이용되는 대표적인 수학도구가 미분방정식이며, 미분방정식을 풀이하기 위하여 이용되는 공학적 도구 중 하나가 유한차분법과 같은 수치해석 기법이다. 기존의 공학적 접근으로, 분포 모형(distributed model)으로 설명되는 문제들이며, 본 연구에서 분류한 기준으로는 물리구조를 갖는 동질소통구조의 문제이다. 토질, 관개, 재료 분야의 문제를 각각 선정하여, 유한차분으로 정식화한 과정을 설명하고, 이를 ASAP에서 구현하고, 모의하였다.

토질분야 문제로는 연직배수를 고려한 1차원 토질압밀을 모의하였다. 토양 내 과잉간극수압을 이용하여, 토질 압밀을 산정하는 모형으로 양단 배수 모형을 적용하여 50일, 95일, 145일, 160일이 지난 시점의 상태를 도시하였다. 참조한 연구결과와 일치하는 결과를 확인하였으며, ASAP의 기본 기능을 통하여 시각화, 그래프 등을 쉽게 획득할 수 있었다.

관개분야 문제로는 점적관개에 의한 토양수분 재분배를 모의하였다. 토양수분 재분배는 mixed-form Richard 식으로 정식화하였으며, van Genuchten 식을 이용하여 수리전도도를 산정하였다. 이층 토양을 모의하여 상층부는 sandy loam, 하층부는 loam으로 설정하여 2.25 l/hr로 4시간 관개 후, 48시간을 모의하였다. 상층부의 수리전도도가 높아서 이층의 접합부분에서

수분이 정체하는 현상을 확인할 수 있었다. ASAP는 시뮬레이션 중간에 값을 변경하여, 변경된 값을 기준으로 시뮬레이션을 계속 진행할 수 있어, 향후 다양한 관개 스케쥴에 따라 토양수분의 재분배를 평가할 수 있을 것으로 기대한다.

재료분야 문제로는 배수갑문의 염해를 모의하였다. 배수갑문의 콘크리트 피어는 항상 해수와 인접하여 염해가 일어난다. 이에 따라염해를 방지하기 위하여 차수제와 같은 보수보강 작업이수행되는데, 효과적인 시공을 위해서 콘크리트 내부의 염화물질 농도를 모의하는 것이 중요하다. ASAP를 이용한 모의실험 결과, SRC-31.4 콘크리트를 이용하여 무보강 시 32년 후 철근 부식이우려되며, 20년에 5cm을 새 콘크리트로 대체 보강하였을 경우, 38년으로 생잔기간을 6년 연장할 수 있었다. ASAP를 이용함으로써, 시나리오 기능을 통한 다양한 보수보강에 따른 비교 평가가 가능한 장점을 취할 수 있다.

이질소통구조를 갖는 문제는 주로 네트워크로 소개되는 문제로써 시스템을 구성하는 요소간에 연결강도, 상관도가 다른 문제로 이해할 수 있다. 이질소통구조를 갖는 문제 중 위상구조를 갖는 문제로 HPAI 감염농가의 확산을 모의한 방재문제, 물리구조를 갖는 문제로 셀룰러 오토마타를 이용하여 트러스를 해석한 구조문제를 제시하였다.

방재분야 문제로는 전염확률을 고려한 감염농가의 확산을 모의하였다. 2008년 전라도 김제시에서 발생하였던 HPAI 감염에 대한 역학조사 결과를 이용하여, 농가의 전염확률을 고려한 감염확산을 모의하였다. 시뮬레이션을 통하여 잠복기를 갖는 전염병의 확산 패턴에서 피크가 지연되어 나타나는 현상을 확인하였으며, 모의실험에서 빠른 시일 내에 감염된 대부분의 농가가 실제 HPAI가 발병했던 농가와 일치하였다.

구조분야 문제로는 트러스 구조해석을 수행하였다. 트러스의 각절점에서 힘의 평형방정식을 수립하고, 부재에 작용하는 미소변형을 내력으로 계산하여 지배방정식을 정식화하여, 컴포넌트를 개발하였다. 개발한 컴포넌트를 이용하여 10-트러스 문제, 5-트러스 문제, 11-트러스 문제를 풀이하여 기존 연구 결과와 동일한결과를 확인하였다. 구조 분야 문제를 통하여 기존 방식에서는 계산과정동안 후처리 기능을 수행할 수 없었으나, ASAP에서는 계산과

동시에 그래프가 반영되어, 강체 운동과 같이 계산오류가 발생할 때, 연구자에게 직관적인 피드백이 가능하였다.

마지막으로 복제기능은 시스템의 형상을 만들어가는 과정을 지원하므로, 초목의 형상을 모의하는 초목성장 모델과 전처리기능으로 활용한 등가트러스 문제를 선정하여 풀이하였다. 복제기능이 지역시스템 분야에서 공학적으로 한정된 모델의 형태로이용하고 있어, 다양한 문제를 제시할 수는 없지만, 진화 알고리즘, 유전 알고리즘과 같이 최적화 메커니즘으로 응용되는 점에 비추어, 시스템 시뮬레이션 측면에서도 시스템의 성장, 진화, 사멸을모의하기 위한 주요한 메커니즘으로 발전할 것으로 예상한다.

본 논문은 시스템 시뮬레이터를 개발하고, 이를 다양한 분야의 문제에 적용함으로써 ASAP의 가능성을 평가하였다. ASAP는 연구자가 쉽게 모형을 시뮬레이션할 수 있는 환경으로, 기존의 수학적/공학적 방식과 다른 접근법을 제공한다. 전체시스템을 이해하고, 이를 세분하는 방식에서 탈피하여, 시스템을 구성하는 작은 요소의 거동을 정의하고, 이들 요소를 연결하여 시뮬레이션을 시도해 봄으로써, 모델의 기능을 추가하고, 보완하여 전체 시스템을 재현하는 방식으로 지역시스템을 이해할 수 있을 것으로 기대한다.

# 5.2 향후 연구과제

본 연구에서 ASAP를 개발하면서, 사용자의 편의성을 향상시키기 위한 전처리 혹은 후처리 기능 구현은 다소 부족하였다. 기존 컴포넌트를 단순히 이용하고, 새롭게 ASAP를 접하는 이용자에게 진입장벽을 낮추기 위해서는 직관적인 GUI를 제공하고, 모델을 구성하는 과정에서 자동화가 가능한 루틴에 대한 전처리가 필요하다. 또한 시뮬레이션을 통하여 주어진 조건에서 결과를 모의할 수 있으나, 최적의 결과를 찾기 위해서 조건을 거꾸로 찾아내는 최적화 도구나 여러 가지 시나리오 조건을 이용하여 산정된 결과를 다각도로 비교할 수 있는 시나리오 비교 도구와 같은 유틸리티가 필요하다. 이러한 기술적 기능 개선을 통하여 사용자는 쉽게 모델을 적용할 수 있을 것으로 기대할 수 있으며, 반복 모의 및 시나리오 환경을 통하여 연구자는 계획을 수립하여 평가하고, 대안을 찾아가는 과정을 수행할 수 있다. 이러한 기술적 기능을 보강함으로써 시스템 모의를 넘어 시스템을 고려한 의사결정자의 판단을 지원할 수 있는 의사지원 도구(Decision Support System, DSS)로 발전할 수 있을 것으로 기대한다. 또한, 사물인터넷(Internet of Things, IoT)와 접목하여 ASAP의 각 컴포넌트는 하나의 사물(things)로 대치할 수 있고, 이를 통하여 ASAP는 가상환경뿐만 아니라, 현실세계에 실존하는 다양한 시스템으로 확장하여 모사하고, 평가할 수 있을 것으로 판단된다.

더불어 지속적인 발전을 위해서는 ASAP를 이용한 모델 및 시뮬레이션 결과를 공유하고, 이를 개선하고, 서로 기여할 수 있는 생태계가 필요하다. 시뮬레이션 결과를 동적으로 손쉽게 재현할 수 있는 웹 기반 시뮬레이션 뷰어와 연구결과를 효과적으로 발표하고, 교환할 수 있는 ASAP 컴포넌트 마켓을 마련함으로써 사용자들이모델을 쉽게 개발하고, 공개할 수 있는 기반을 제공할 수 있다. 이러한 생태계를 통하여 신규 사용자의 진입을 장려하고, 기존 사용자 경험(User Experience, UX)을 진작하여 사용자 저변을 확보하고, 관련 분야의 인력 교육 및 연구역량 계발을 지원하여 지역시스템공학의 발전에 이바지할 수 있기를 기대한다.

# 참고 문헌

- 1. Acock, B., and Reddy, V. R. (1997). "Designing an object-oriented structure for crop models." *Ecological Modelling*, 94(1), 33-44.
- 2. Bahill, A. T., and Gissing, B. (1998). "Re-evaluating systems engineering concepts using systems thinking." *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 28(4), 516-527.
- 3. Beck, F., and Diehl, S. (2011). "On the congruence of modularity and code coupling." *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, ACM, Szeged, Hungary, 354-364.
- 4. Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). "A Manifesto for Agile Software Developers." <a href="http://agilemanifesto.org/">http://agilemanifesto.org/</a> >. (10 June, 2014).
- 5. Bhatti, M. A. (2005). Fundamental finite element analysis and applications: with Mathematica and Matlab computations, Wiley.
- 6. Bloch, J. (2008). Effective Java, Pearson Education.
- 7. Bro-Nielsen, M. (1998). "Finite element modeling in surgery simulation." *Proceedings of the IEEE*, 86(3), 490-503.
- 8. Celia, M. A., Bouloutas, E. T., and Zarba, R. L. (1990). "A general mass-conservative numerical solution for the unsaturated flow equation." *Water Resources Research*, 26(7), 1483–1496.
- 9. Chandrupatla, T. R., and Belegunda, A. D. (2002). *Introduction to finite elements in engineering*, Prentice Hall.
- 10. Collier, N. (2003). "Repast: An extensible framework for agent simulation." *The University of Chicago's Social Science Research*, 36, 2003.
- 11. Cormen, T. H. (2009). Introduction to Algorithms, Mit Press.
- 12. Cross, H. (1930). "Analysis of Continuous Frames by Distributing Fixed-End Moments." *Proceedings of the*

- American Society of Civil Engineers (ASCE), 919–928.
- 13. Doorenbos, J., and Pruitt, W. O. (1977). "Guidelines for Predicting Crop Water Requirements." *AO Irrigation and Drainage Paper 24*, FAO, Rome.
- 14. Eberlein, R. L., and Peterson, D. W. (1992). "Understanding models with Vensim<sup>TM</sup>." *European Journal of Operational Research*, 59(1), 216-219.
- 15. Emrich, S., Suslov, S., and Judex, F. "Fully Agent-Based Modelling of Epidemic Spread Using AnyLogic." *Proc., Proc. EUROSIM*, 9-13.
- 16. Englander, R. (1997). *Developing Java Beans*, O'Reilly Media, Incorporated.
- 17. Fielding, R. T. (2000). "Architectural styles and the design of network-based software architectures." University of California, Irvine.
- 18. Forrester, J. W. (2007). "System dynamics—a personal view of the first fifty years." *System Dynamics Review*, 23(2-3), 345-358.
- 19. Gürdal, Z., and Brian, T. (2000). "Cellular Automata for Design of Truss Structures with Linear and Nonlinear Response." AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference and exhibit, AIAA, Atlanta, Georgia.
- 20. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software, Pearson Education.
- 21. Gardner, M. (1970). "Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life"." *Scientific American*, 223, 120-123.
- 22. Goodliffe, P. (2007). Code Craft: The Practice of Writing Excellent Code, No Starch Press.
- 23. Han, Y., Goetz, S. J., Lee, J., and Yoon, S. (2012). "Simulating Network Structures Using Bernoulli'S Principle." *Advances in Complex Systems*, 15(05).
- 24. Hazelrigg, G. A. (1996). Systems engineering: an approach to information-based design, Prentice Hall.
- 25. Jones, J. W., Keating, B. A., and Porter, C. H. (2001). "Approaches to modular model development." *Agricultural Systems*, 70(2–3), 421–443.
- 26. Kehtarnavaz, N., and Gope, C. "DSP System Design Using Labview and Simulink: A Comparative Evaluation." *Proc.*,

- Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, II-II.
- 27. Klügl, F., Herrler, R., and Fehler, M. "SeSAm: implementation of agent-based simulation using visual programming." *Proc., Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, ACM, 1439–1440.
- 28. Kreyszig, E. (2010). Advanced Engineering Mathematics, John Wiley & Sons.
- 29. Lindenmayer, A. (1968). "Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs." *Journal of Theoretical Biology*, 18(3), 300-315.
- 30. Maage, M., Helland, S., Poulsen, E., Vennesland, O., and Carsen, J. E. (1996). Service life prediction of existing concrete structures exposed to marine environment, American Concrete Institute, Farmington Hills, MI, ETATS—UNIS.
- 31. Macal, C. M., and North, M. J. (2010). "Tutorial on agent-based modelling and simulation." *J of Sim*, 4(3), 151-162.
- 32. Matthews, R. (2013). "Structsource: Moment Distribution Example."

  <a href="http://www.structsource.com/analysis/types/momentdistributionexample.html">http://www.structsource.com/analysis/types/momentdistributionexample.html</a>.
- 33. McConnell, S. (2004). Code complete, Microsoft Press.
- 34. Mualem, Y. (1976). "A new model for predicting the hydraulic conductivity of unsaturated porous media." *Water Resources Research*, 12(3), 513-522.
- 35. O'reilly, T. (2007). "What is Web 2.0: Design patterns and business models for the next generation of software." *Communications and Strategies*, 65(1), 17-37.
- 36. Peierls, T., Goetz, B., Bloch, J., Bowbeer, J., Lea, D., and Holmes, D. (2006). *Java Concurrency in Practice*, Pearson Education.
- 37. Prusinkiewicz, P. (1998). "Modeling of spatial structure and development of plants: a review." *Scientia Horticulturae*, 74 (1-2), 113-149.
- 38. Reynolds, C. W. (1987). "Flocks, herds and schools: A distributed behavioral model." *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM, 25–34.

- 39. Richmond, B. (1994). "Systems thinking/system dynamics: Let's just get on with it." *System Dynamics Review*, 10(2-3), 135-157.
- 40. Sadegh Zadeh, K. (2011). "A mass-conservative switching algorithm for modeling fluid flow in variably saturated porous media." *Journal of Computational Physics*, 230(3), 664-679.
- 41. Santé, I., García, A. M., Miranda, D., and Crecente, R. (2010). "Cellular automata models for the simulation of real-world urban processes: A review and analysis." *Landscape and Urban Planning*, 96(2), 108-122.
- 42. Tatting, B., and Gürdal, Z. "Cellular automata for design of two-dimensional continuum structures." *Proc., Proceedings of 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.*
- 43. Terzaghi, K. (1943). *Theoretical soil mechanics*, J. Wiley and Sons Chapman and Hall, New York, London,.
- 44. Tisue, S., and Wilensky, U. "Netlogo: A simple environment for modeling complexity." *Proc., International Conference on Complex Systems*, 16-21.
- 45. University of Alberta (2001). "University of Alberta ANSYS Tutorials: Two Dimensional Truss." <a href="http://www.mece.ualberta.ca/tutorials/ansys/BT/Truss/Truss.html">http://www.mece.ualberta.ca/tutorials/ansys/BT/Truss/Truss.html</a> >. (10 Nov., 2013).
- 46. Van Evert, F. K., and Campbell, G. S. (1994). "CropSyst: A Collection of Object-Oriented Simulation Models of Agricultural Systems." *Agron. J.*, 86(2), 325-331.
- 47. van Genuchten, M. T. (1980). "A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils1." *Soil Sci. Soc. Am. J.*, 44(5), 892-898.
- 48. Whitley, K. N., and Blackwell, A. F. (2001). "Visual Programming in the Wild: A Survey of LabVIEW Programmers." *Journal of Visual Languages & Computing*, 12(4), 435–472.
- 49. Williams, F. W. (1973). "Comparison between sparse stiffness matrix and sub-structure methods." *International Journal for Numerical Methods in Engineering*, 5(3), 383-394.
- 50. Wolfram, S. (1983). "Statistical mechanics of cellular automata." *Reviews of Modern Physics*, 55(3), 601–644.
- 51. Wolfram, S. (1984). "Universality and complexity in cellular automata." *Physica D: Nonlinear Phenomena*, 10(1–2), 1–35.

- 52. Yi, H. (2003). "Simplifying complexity and emerging self-organization of integrated agricultural system models." Ph.D. dissertation, Seoul National University.
- 54. 김시환, 윤성수, 박진선, and 전정배 (2011). "개선된 점진적 구조 최적화 기법을 이용한 콘크리트 구조물의 응력경로 탐색 " 한국농 공학회논문집 53(6), 153-163.
- 55. 김윤순, 정남수, 이정재, and 윤성수 (2004). "점진적 최적화 기법 에서 불규칙 삼각망을 이용한 평면구조의 응력경로 탐색모델의 개발 " 한국농공학회논문집 46(4), 65-71.
- 56. 김태곤 (2007). "컴포넌트 속성의 집중관리 및 물질전달계와 정보 전달계의 분리를 통한 동적시스템 시뮬레이터의 개선." M.S. thesis, Seoul National University.
- 57. 김태곤, and 이정재 (2007). "정보 전달계의 분리를 통한 GASS 의 개선 " *한국농공학회논문집* 49(1), 101-109.
- 58. 김현영, and 박승우 (1988). "관개용 저수지의 일별유입량과 방류 량의 모의발생(I)-선형 저수지 모형에 의한 유입량의 추정-." 한 국농공학회지, 30(1), 50-62.
- 59. 남원호, and 최진용 (2013). "실시간 저수위 및 용수공급 취약성 지표를 활용한 농업용 저수지 운영 기준 개발." *한국농공학회논문* 집, 55(6), 77-85.
- 60. 노희명, 권재필, 김승현, and 류순호 (2000). "작물 흡수를 고려한 3차원 토양수분 분포 모델 개발을 통한 최적 점적 관개 연구 Cropped Conditions." 한국농화학회지, 43(2), 116-123.
- 61. 농림수산식품부 (2012). 고병원성 조류인플루엔자(HPAI)의 유입 및 전파확산경로 예측을 위한 가금 산업의 유통 감시 네트워크 시 스템 개발, 농림수산식품부.
- 62. 농림수산식품부, 국립수의과학검역원, and 가축위생방역지원본부 (2008). (고병원성)조류 인플루엔자 백서 2008, 농림수산식품부.
- 63. 다음 (2014). "다음 지도." <a href="http://map.daum.net">http://map.daum.net</a> >. (13 Jun., 2014).
- 64. 루흐알라, 엘. (2012). *누구나 생물*, Gbrain, 서울.
- 65. 박미정, 허현, 김태곤, 서교, and 이정재 (2006). "크루스칼 알고 리즘과 불규칙 삼각망을 이용한 유역 추출 알고리즘." *한국농공학 회논문집*, 48(4), 3-12.
- 66. 배연정 (2001). "농업수리구조물 통합설계 시스템을 위한 PC 모델의 적용과 개선." 서울대학교 대학원.
- 67. 봉태호, 손영환, 노수각, and 박재성 (2012). "압밀계수의 공간변

- 동성에 따른 압밀도의 확률론적 해석." *한국농공학회논문집*, 54(3), 55-63.
- 68. 서교, 이정재, 김태곤, and 이호재 (2005). "GASS를 이용한 농산 물 유통 시스템 시뮬레이션." *한국농공학회논문집*, 47(4), 3-12.
- 69. 손혜영 (2012). *LabVIEW의 정석 -기본편*, 인피니티북스, 경기 도 고양.
- 70. 유승환, 최진용, and 장민원 (2006). "논벼에 대한 Penman-Monteith와 FAO Modified Penman 공식의 작물 계수 산정." *한* 국농공학회논문집, 48(1), 13-23.
- 71. 윤득노, 김태곤, 한이철, and 이정재 (2010). "PERT 요소공정의 주경로 확률 산정 모델 개발." *한국농공학회논문집*, 52(2), 27-34.
- 72. 윤성수, and 이정재 (2002). "점진적 구조 최적화 기법을 응용한 철근콘크리트 부재의 배근." *한국농공학회지*, 44(2), 127-135.
- 73. 윤용남 (2007). *수문학:기초와 응용*, 청문각.
- 74. 윤하정, 박지용, 박최규, 이윤정, 하진경, 정승교, 이용주, 정지원, 이수두, 권준헌, and 문운경 (2008). "'03/'04 및 '06/'07 H5N1 고병원성 조류인플루엔자 유행의 역학적 특성." 한국예방수의학회 (구 한국수의공중보건학회).
- 75. 이성용, 김태곤, and 이정재 (2009). "연속체의 이산화에 의한 등 가트러스모델 개발." *한국농공학회논문집*, 51(3), 45-52.
- 76. 이정재 (1991). "段階別 塑性解析 技法을 利用한 뼈대構造의 信賴 性 解析 모델 開發." Ph.D. dissertation, 서울대학교.
- 77. 이한귀, 김태곤, and 이정재 (2012). "요소 독립성이 유지되는 반복해법에 의한 객체지향 유한요소모델 개발 "*한국농공학회논문집*, 54(2). 115-125.
- 78. 이형진 (2014). "감마분포로 추출한 허브 네트워크와 확산 및 구조 신뢰성 해석." Ph.D. dissertation, 서울대학교.
- 79. 임종수 (2013). (임종수의) Simulink 8. 높이깊이.
- 80. 최순군, 최진용, 남원호, 허승오, 김학진, 정선옥, and 한경화 (2012). "점적관개에 따른 토양수분 재분배 균일성 평가." 한국농 공학회논문집, 54(3), 19-28.
- 81. 최원 (2012). "Regular and irregular truss structure models equivalent to continuum structure." Ph.D. dissertation, Seoul National University.
- 82. 한이철 (2012). "Analysis of the socio-geometric systems and continuum structures using complex network theory." Ph.D. dissertation, 서울대학교.

# **Abstract**

# Development of Agricultural Systems Application Platform enhancing Connection, Communication, and Cloning

Taegon Kim
Major in Rural Systems Engineering
Department of Landscape Architecture and
Rural Systems Engineering
The Graduate School
Seoul National University

System simulation consists of complex processes to model component behaviors and describe component interactions. The complexity of these processes prohibits a system simulation model from establishing the relationships between upper—models and sub—models. In this study, we replace the procedural paradigm with an object—oriented paradigm for complex system simulation to reduce the complexity. Many system simulation studies have adopted an object oriented paradigm, but the integration parts still have procedural codes even though these studies have attempted to mitigate dependency and complexity issues.

System simulation tools have been reviewed to extract key features for simulating systems. The Unidirectional Information Fetch (UIF) method was especially selected as the key feature referenced from the Generic Agricultural Systems Simulator (GASS). We also examined the behaviors of living organisms which is a complex and well-organized system for finding the clues of the generic system simulation. The living organisms has three major behavioral features: connection, communication, and cloning. First, organs are connected with not only adjacent elements but also long-distance entities for information exchange. Second, the interaction among organs is possible through various means. Third, the birth and death of elements evolve into system structures through cloning.

In this study, the Agricultural Systems Application Platform (ASAP) was developed by enhancing the three key features: connection, communication, and cloning. "Connection" was used to represent geometric and topologic links based on network theories. "Communication" was incorporated to deal with both homogeneous and heterogeneous structures. "Cloning" was added to the ASAP for changes to system boundary conditions. Additional utilities and protocols were also provided to model developers as the user interface and guidelines for the simulation environment setup and model components. We applied ASAP to nine rural systems problems for evaluating its applicability. The ASAP provided similar simulation outputs compared to common complex models results using a much simpler model structure

The ASAP will be utilized for simulating engineering problems with a bottom-up modelling approach instead of a conventional top-down approach. Through continuous future development and updates, the ASAP can be a platform for sharing the deployment and knowledge of rural and agricultural systems among engineers.

Keywords: Connection, Communication, Cloning, Object oriented paradigm, System simulation, Platform

Student Number: 2007-30313