The 2nd Annual Conference of the Korean Society for Bioinformatics (KSBI2003)

Development of Bio-Inspired System Software Architecture Based on Multi-Agents Framework[1)]

멀티 에이전트 프레임워크 기반의 생태계 모방 시스템 소프트웨어 아키텍처 개발

Hakjoo Lee, Jiyoung Song, KeunJun Lee, Sungyong Park, Sungwon Jung, Jihoon Yang, Jongho Nang

Department of Computer Science, Sogang University, Seoul, Korea
E-mail: or4nge@sogang.ac.kr, parksy@ccs.sogang.ac.kr, and jungsung@ccs.sogang.ac.kr

Abstract

네트워크의 발전으로 인해 현재 존재하는 소프트웨어 구조에 몇 가지 문제점을 내포하게 되었다. 이러한 환경에 대응하기 위해 생태계 구조를 모방한(Bio-inspired) 네트워크 기반 적응 생존형 시스템을 제안한다. 여기서는 생태계의 여러 특성 중 적응성(adaptability) 확장성(scalability), 생존성(survivability)을 모델링 하고자한다. 이 시스템은 상기의 특성을 포함하는 몇 개의 계층으로 구성되어 있다. 독립된 개체의 역할을 하는 에코전트 레이어와 에코전트의 활동을 지원하는 에코전트 플랫폼 레이어, 효율적인 네트워크 활용을 위한 플랫폼 콜레버레이션 레이어로 이루어져 있다. 본 논문에서는 이러한 시스템의 구체적인 기능과 구성 그리고 이 시스템의 활용 분야에 대해 살펴본다.

1. Introduction

1990년대부터 형성된 인터넷 붐으로 어느 곳에서도 전 세계를 연결할 수 있는 네트워크 인프라가 구축되었고 무수한 정보의 공유가 가능하게 되었다. 이러한 구조에 맞는 응용 서비스들을 개발하기 위해서는 단순히 네트워크 인프라 처리 속도를 향상시키는 것만으로는 효율적인 해결 방안이 될 수 없으며 현재의 네트워크 구조나 소프트웨어 구조가 갖고 있는 여러 가지 문제점을 해결해야 한다. 즉, 현존하는 응용 서비스의 소프트웨어 구조는 양적으로 엄청나게 증가하는 네트워크의 호스트의 수를 감당할 수 있는 확장성(scalability)에 문제가 있으며, 다양한 환경 변화에 대한 적응성(adatability)이나 결함에 대처할 수 있는 가용성(availability) 측면에서도 많은 문제점을 내포하고 있다. 이러한 문제점의 해결 방안을 생태계에서 찾고자 한다. 생태계의 구조를 모방한 (Bio-inspired) 네트워크 기반 적응 생존형 생태계 모방 시스템 소프트웨어를 개발함으로써 이러한 문제점들이 해결될 것이라고 생각한다.

생태계 모방 시스템 소프트웨어는 멀티 에이전트 기반의 소프트웨어 구조가 가장 적합하다. 소프트웨어 에이전트는 자율성, 협동성, 적응성을 갖고 사람의 노력과 시간이 필요한 여러 가지 일들을 보다 쉽고 편리하게 대행해주는 소프트웨어이다. 이러한 특징은 생태계의 현상을 적절히 모델링 하는데 매우 적합하다. 여기에 생태계 모방 시

1) 본 논문은 2003년도 산업자원부 차세대신기술개발사업의 지원을 받아 수행되었음.

스템을 위한 시스템 소프트웨어 에이전트의 개념에 진화, 적응, 생존 등의 생태계 현상을 적용할 수 있다. 이러한 목적으로 벌의 생태를 모태로 하여 에코전트(eco-gent, ecological+agent) 및 플랫폼을 개발하게 되었다.

2. 생태계 특성 및 적용

생태계는 중앙의 특정한 관리 없이도 자율적인 개체의 판단에 의해 군집의 기능을 유지하고 발전시킨다. 이러한 특성은 현재 인간과 인간이 만든 대부분의 전자 장치들이 연결되어가는 네트워크 환경에 매력적인 구조이다. 따라서 생태계의 특징을 살펴보고 이를 적용할 수 있는 방안을 살펴보고자 한다.

2.1 생태계의 특성

생태계 모방에 대한 연구는 생물학자들의 개미 군집에 대한 연구 이후로 관심을 갖게 되었다. 논리적인 사고 과정을 거치지 않는 개미들이 실질적으로는 상당히 복잡한 대화 전달 체계와 구조를 가지고 있음을 발견하게 되었다. 개미의 유전자나 뇌 조직에는 계획이나 조직성, 제어 기능이 존재하지 않지만 자발적인 조직화 과정을 통하여 문제를 해결한다[1]. 자발적인 조직화의 특징은 어떤 제어나 절차의 필요성 없이 개체의 협력을 제공한다. 이러한 생태계의 특징은 네트워크 환경의 궁극적인 목표가 될 수 있다.

여기서는 생태계의 군집 중 대표적인 벌을 관찰해 본다. 벌 군집은 중앙 집중화된 관리 없이 모든 벌 개체가 독립적으로 행동한다. 따라서 지역적인 환경과 다른 벌과의 부분적인 정보교환을 근거로 판단하고 움직이게 된다. 즉 다양한 환경 변화에 대한 적응성(adaptability)을 가지게 된다. 환경에 적응된 벌 군집은 벌의 수를 늘려 확장을 꾀하게 된다. 이때 단순한 개체 수의 증가 외에도 우수한 개체로 진화를 유도하게 된다. 즉 필요한 개체의 확장성(scalability)을 가지고 있다. 벌은 천적으로부터 많은 수의 벌이 공격을 받더라도 여전히 벌 군집은 생존할 수 있다. 이는 어떤 하나의 벌에 군집의 생존을 의지하는 것이 아니기 때문이다. 즉 결함에 대처할 수 있는 생존성(survivability)을 가지고 있다.

2.2 생태계의 특성 적용

2.2.1 적응성(adaptability)

생태계는 인위적인 도움 없이 자체적인 성장과 조직화를 꾀한다. 또한 이러한 성장은 항상 안정적인 개체 수 유지를 위한 방향으로 진행된다. 개체 간의 발생하는 행동과 관계를 통하여 이질적이고 동적인 생태계 환경에서 개체의 활동은 유지되어야 한다. 즉 자신의 활동이 활발한 개체는 복제(replication)와 같은 방법을 이용하여 개체의 수를 늘릴 수 있다. 이때 에너지 개념을 이용하여 적용 가능하다. 즉 많은 활동을 하는 개체는 다량의 에너지를 확보하게 되고 이를 근거로 새로운 개체를 생성시킬 수 있다. 반대로 에너지를 근거로 소멸 가능하다[2]. 또한 개체 간의 이주(migration)를 적용할 수 있다. 현재 환경이 개체에 적합하지 않으면 다른 환경으로의 이주를 유도하고 가능하게 한다.

2.2.2 확장성(scalability)

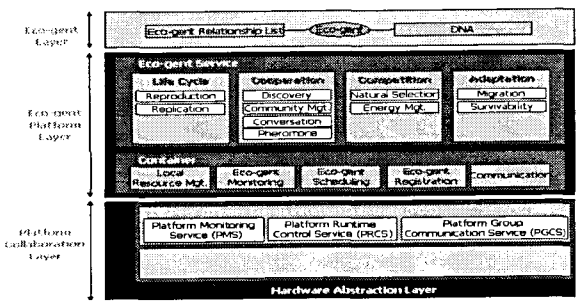
개체는 독립적으로 행동한다. 따라서 각 개체의 수의 증가는 수행 가능한 능력의 증가를 의미한다[3]. 또한 능률의 향상은 개체 수에 대한 상대적인 수행 능력 향상으로 평가 된다. 또 개체들은 타 개체 간의 협력 모델 형성을 꾀함으로써 비 중앙 집중적인 구조의 한계를 극복할 수 있다. 이러한 개체의 생성은 적합도(fitness)와 에너지를 근거로 하여 자기 복제(replication)와 생식(reproduction)을 이용한다.

2.2.3 생존성(survivability)

생태계는 교배와 돌연변이를 통해 다양한 개체의 생성과 진화된 개체가 선택되는 과정을 반복하게 된다. 이러한 과정이 환경의 변화에 대한 생존 가능성을 높이게 된다. 즉 진화에 의한 자연 선택은 개체의 에너지가 소진 되었을 때의 소멸과 충분한 에너지를 가진 개체의 복제나 교배로 이루어진다.

3. 생태계 모방 시스템 소프트웨어 설계

생태계 모방 시스템 소프트웨어는 분산 환경에 보다 적합한 피어-투-피어 통신 구조를 기반으로, 생태계의 협동과 경쟁, 진화·적응 모델을 적용하여, 생존성과 적응성, 확장성을 높인 이동 에이전트 시스템을 목표로 한다.



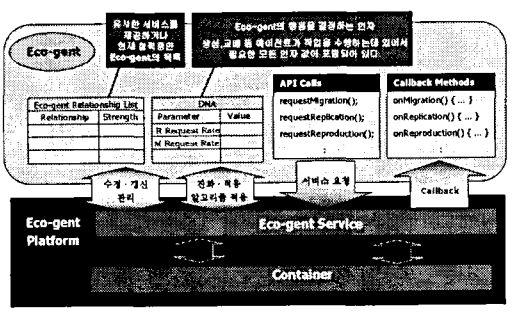
[그림 1] 플랫폼 아키텍처

생태계 모방 시스템 소프트웨어는 [그림 1]에 나타나있는 것과 같이 3층의 레이어(layer)로 구성되어 있다. 최상층은 에코전트 레이어(Eco-gent layer)로 에이전트 소프트웨어에 해당한다. 에코전트는 일반적인 에이전트와 유사하지만, 다른 에이전트들과 달리 진화·적응에 적합한 구조를 가지고 있다. 두 번째, 에코전트 플랫폼 레이어(Eco-gent platform layer)는 에코전트의 실행 환경(execution environment)을 위한 서비스들을 제공한다. 플랫폼 서비스는 4개의 컴포넌트로 구성되어있다. 마지막으로 플랫폼 콜래버레이션 레이어는(Platform collaboration layer)은 에코전트 시스템의 저수준 서비스와 플랫폼 간의 통신을 담당하게 된다. 플랫폼 콜래버레이션 레이어에서는 JXTA[6]를 사용, 분산 피어-투-피어(Peer-to-peer) 환경에서 에코전트가 수행될 수 있도록 한다.

3.1 에코전트 레이어

에코전트는 에코전트 인터페이스를 상속받아 사용자가 에이전트를 설계할 수 있다. 에코전트는 기본적으로 릴레이션쉽 리스트(relationship list)와 DNA 리스트를 갖게 된다.([그림 2] 참조) 릴레이션쉽 리스트는 연관관계가 있는 에코전트들에 대한 리스트로, 키워드를 통한 디스커버리를 하거나 다른 새로운 에코전트를 찾을 때 사용된다. DNA 리스트는 새로운 에코전트를 생성하여 진화하거나, 환경이나 사용자의 요구에 적응하기 위한 기본 파라미터를 갖고 있어, 이에 따라 에코전트의 행동이 결정된다. 이런 DNA 리스트는 확장 가능한 요소이다.

에코전트는 필요한 서비스를 에코전트 플랫폼 레이어에 요청하고, 서비스에 대한 콜백을 받는 방식으로 동작한다. 예를 들어, 새로운 에코전트 플랫폼으로 이동하기 위해서 requestMigration()을 호출하여, 에코전트 플랫폼 레이어에 서비스를 요청한다. 에코전트 플랫폼 레이어에서는 에코전트가 다른 플랫폼으로 이동하기 위한 준비를 하고, 이를 완료하면, 에코전트 콜백 함수인 onMigration()을 호출하게 된다.



[그림 2] 에코전트 설계

3.2 에코전트 플랫폼 레이어

에코전트 플랫폼 레이어는 에코전트 플랫폼 서비스와 컨테이너로 구성되어 있다.

3.2.1 에코전트 플랫폼 서비스

에코전트 플랫폼 서비스는 4가지의 컴포넌트로 구성되어 있다. 에코전트가 원활한 활동을 할 수 있는 환경을 마련해주는 역할을 한다. 구체적인 컴포넌트로는 라이프 사이클 컴포넌트, 코퍼레이션 컴포넌트, 컴피티션 컴포넌트, 어뎁테이션 컴포넌트가 있다.

3.2.1.1 라이프 사이클 컴포넌트

라이프 사이클 컴포넌트는 에코전트의 번식과 기본 생성을 제공하는 서비스이다. 내부적인로 리프로덕션 서비스와 리플리케이션 서비스로 구성되어있다.

리프로덕션 서비스는 진화 알고리즘을 적용하여 불규칙적으로 변화된 다양한 에코전트를 생성시켜 현재 상황에 더 적합한 에코전트를 생성시키기 위하여 필요한 서비스이다. 각각의 에코전트들은 스스로의 필요에 따라 서비스에 번식을 요청하게 되며, 서비스는 새로운 개체가 부모로부터 에너지를 충당하므로 번식이 가능한 에너지 레벨을 가지고 있는지 확인하여 해당 에코전트를 리프로덕션 대기 풀에 삽입한다. 교배와 변이는 풀에 등록된 여러 가지 에코전트들 중에서 적합도가 가장 높은 에코전트 두 개를 선택하여 이루어진다. 교차(crossover)과정은 에코전트 설계자의 선택에 따라 개미군집 최적화 알고리즘과 진화 알고리즘을 통해서 이루어진다. 기본적으로 해당 에코전트의 속성을 변화시키는 것이 그 목적이다. 한 번 교배가 이뤄지면 2개의 새 개체를 생성된다. 이때 정해진 확률값을 가지는 돌연변이율이 적용된다.

리프로덕션 서비스의 역할은 첫째, 진화 알고리즘을 적용하여 변화된 다양한 개체를 생성시키는 것이다. 둘째, 각 개체로부터 요청이 있을 때 개체 수를 늘이기 위하여 사용된다. 셋째, 현재 환경에 적합한 개체를 생성시키고 증가시키는 역할을 한다.

리플리케이션 서비스는 자신과 같은 개체를 하나 더 생성시키는 것을 목적으로 하는 서비스로, 두 가지 경우에 일어나는데, 하나는 에코전트가 스스로 자신의 적합도가 비교적 높다고 판단하여 자신처럼 적합도가 높은 에코전트의 양을 늘리기 위해 요청하는 경우이고, 다른 하나는 이동시에 에러가 발생할 것에 대비해 에코전트를 백업하는 경우이다. 후자의 경우는 무조건 자신과 완전히 같은 에코전트를 하나 생성하는 것이며, 전자의 경우는 서비스에서 해당 에코전트가 복제에 충분한 에너지를 가지고 있는지 검사하고, 검사를 통과할 경우 요청한 에코전트의 복사본을 만들고 마이그레이션 서비스만을 거친 새로운 에코전트를 생성하게 된다.

리플리케이션 서비스의 역할은 특정 개체를 복제할 때 사용되며 적합도가 높은 개체의 수를 안전하게 증가시키는 것이다. 따라서 특정 개체의 활동량이 많을 때의 혼잡을 사전에 예방할 수 있거나 새로운 개체의 생성으로 집중화를 막고 로드 균형(load balancing)을 유지시킬 수 있다. 이때 리프로덕션과의 근본적인 차이는 교배의 사용 유무이다.

리플리케이션의 동작 순서는 우선 부모가 될 개체가 높은 적합도(fitness)와 충분한 에너지를 가지고 있는지 검사한다. 높은 적합도와 충분한 에너지를 가지고 있다면 자신의 복제를 허용한다. 리프로덕션과 마찬가지로 교배 중에는 지정된 돌연변이율이 적용된다.

해당 개체가 2배씩 늘어나는 것이므로 이에 대한 개체간의 통신량도 증가되고 자원소비와 복잡도(complexity)의 증가의 부정적인 면이 존재하게 된다. 그러나 지나친 자원소비와 복잡도의 증가는 자연선택(natural selection)으로 억제가능하다.

3.2.1.2 코퍼레이션 컴포넌트

코퍼레이션 컴포넌트는 다른 에코전트와 협력하기 위한 컴포넌트이다. 내부적으로 디스커버리 서비스, 커뮤니티 서비스, 컨버

세이션 서비스, 페르몬 서비스로 구성되어 있다.

디스커버리 서비스는 하나의 에코전트가 다른 에코전트와의 협력을 위해 에코전트를 찾을 때 사용된다. 하나의 에코전트가 이름, 서비스 종류, 키워드 등을 통해 로컬과 리모트에 있는 조건에 맞는 에코전트를 찾을 수 있게 된다. 에코전트의 검색 조건을 디스커버리 서비스가 받게 된다. 이러한 조건은 에코전트의 이름, 서비스 종류, 키워드 등의 값이 될 수 있다. 이때 동일한 플랫폼 내의 에코전트를 찾는 경우는 레지스트레이션 컴포넌트를 참조하여 해당 에코전트를 찾게 되고, 다른 플랫폼 상의 에코전트를 찾는 경우는 협동 계층(collaboration layer)의 디스커버리 서비스를 이용하여 이웃한 플랫폼의 에코전트를 찾게 된다.

커뮤니티 매니지먼트 서비스는 두 에코전트가 일 방향 혹은 양 방향의 관계의 사회적인 네트워크를 형성할 수 있도록 관계 리스트를 관리하게 해 주는 서비스이다. 에코전트들은 서로간의 서비스를 제공하고 받음으로써 리스트를 형성할 수 있다. 이러한 리스트를 관리해 주고 원격의 에코전트의 상태와 위치를 모니터링 하여 변화를 반영해주는 기능을 한다. 각 관계 리스트에는 강도(strength)가 포함되어 있어 두 에코전트의 관계의 친밀도를 나타내게 된다. 강도는 서로간의 통신이나 서비스의 요청이 없으면 감소하고, 둘 간의 접촉이 많게 되면 증가하게 된다.

컨버세이션 서비스는 에코전트간의 대화시 사용된다. 이때 ACL(Agent Communication Language) 형태로 메시지를 주고받게 된다. 메시지를 보내고자 하는 에코전트는 메시지의 내용과 발신자 및 수신자들의 정보를 오브젝트 형태로 컨버세이션 서비스에 전달하게 되고, 컨버세이션 서비스는 이것을 ACL 메시지 형식으로 구성하여 컨테이너 계층의 커뮤니케이션 컴포넌트에 넘겨주게 된다. 이 메시지는 협동 계층(collaboration layer)을 통해 해당 플랫폼의 컨테이너 계층에 전달되어 최종적으로 에코전트에게 전달된다. 에코전트는 자신의 메시지 박스(message box)로부터 메시지를 하나씩 가져오게 된다. 메시지 박스는 컨테이너 계층의 커뮤니케이션 컴포넌트가 유지한다.

페르몬 서비스는 생태계의 페르몬을 모방한 서비스이다. 즉 에코전트가 플랫폼에 페르몬을 남길 수 있고 이를 다른 에코전트가 확인하게 된다. 컨버세이션 서비스와의 차이는 컨버세이션 서비스는 특정 대상에게만 메시지를 보내고자 하는 반면 이 페르몬 서비스는 에코전트 스스로 메시지의 확인을 능동적으로 유도한다는 것이다. 또한 페르몬의 경우는 로컬 및 인접한 이웃 플랫폼의 모든 에코전트가 확인 가능하다. 그리고 메시지와는 달리 페르몬은 시간이 지남에 따라 그 강도가 점차 약해져 결국 사라지게 되는 생태적 특성을 반영하고 있다.

3.2.1.3 컴피티션 컴포넌트

컴피티션 컴포넌트는 에코전트의 경쟁으로 자연 선택을 유도하는 서비스로 네츄럴 셀렉션 서비스와 에너지 매니지먼트 서비스로 구성되어 있다.

네츄럴 셀렉션 서비스는 에코전트들의 과다번식 등으로 인해 시스템 자원 용량 한계를 초과하지 못하도록 에코전트들의 수를 적절히 조절하는 역할을 한다. 물론 이 서비스가 존재하지 않더라도 자체적인 시스템 리소스의 한계로 인해 비슷한 결과가 나올 수 있으나, 에코전트 수를 유지시키는 방법(policy)을 직접 조절하고, 플랫폼 자체가 영향을 받을 수 있는 에코전트 포화 상태 및 급격한 자연 선택 현상을 방지하고 적극적으로 대처하기 위해 필요한 서비스라고 할 수 있다.

네츄럴 셀렉션은 구체적으로 두 가지 일을 하는데, 우선 에너지를 리소스로 바꾸는 에너지 단위비용(energy unit cost)을 현재 플랫폼의 가용 리소스에 따라 실시간으로 변화시켜 리소스가 부족해지면 에너지 단위 비용을 높이는 방법으로 리소스 고갈을 막고 이를 이용해 능력이 떨어지는 에코전트를 도태시킨다. 또한 극단적인 상황 즉, 에코전트가 많고 리소스가 부족하여 전체 시

스템이 정체되어있는 상황에서 에코전트들을 강제로 종료시키는 일을 수행할 수 있다.

궁극적으로는 생태계의 개체 수가 임계 한도를 넘어서면 먹이의 부족으로 급격하게 개체의 수가 감소되는 것을 모방한 것으로 시스템의 정체 위기 전에 에코전트의 수를 조절할 수가 있다. 따라서 기본적으로 시스템 환경에 따라서 리소스와 에너지 단위 비용을 적절히 결정하는 것을 전제로 하고 있다. 또한 에코전트의 수를 감소시킬 시점과 이 과정에 대한 변수로써 리프로덕션과 리플리케이션의 비율에 대한 결정이 필요하게 된다.

네츄럴 셀렉션의 과정은 다양한 속성을 가진 에코전트가 다수 존재한다고 가정할 때 시작된다. 에너지를 소진한 에코전트가 있을 경우에는 해당 에코전트를 소멸시킨다. 반면 에너지가 충분하고 적합도가 높은 에코전트들은 현재 플랫폼 상황에 적절하다고 판단되어 리플리케이션 또는 리프로덕션을 하도록 유도된다.

에너지 메니지먼트 서비스는 동작중인 모든 에코전트들의 에너지 보유량을 관리하며, 에너지 단위 비용에 따라 에코전트로부터 에너지를 받고 리소스를 제공하는 역할을 담당한다. 에너지 단위비용은 네츄럴 셀렉션 서비스가 실시간으로 결정해주는 것을 사용하게 된다. 에코전트 또는 타 서비스의 요청에 의해 현재의 에너지 단위 비용 정보를 제공해주며, 타 서비스의 요청에 의해 특정 에코전트가 가진 에너지양을 알려주며, 에코전트의 요청에 따라 자신이 가진 에너지를 타 에코전트에게 넘겨주는 기능을 제공한다.

3.2.1.4 어댑테이션 컴포넌트

어뎁데이션 컴포넌트는 적응성을 위한 서비스로 마이그레이션 서비스와 서바이버빌리티서비스로 구성되어 있다.

마이그레이션 서비스에서는 에코전트의 마이그레이션에 필요한 이동 경로의 설정 및 필요한 코드의 이동에 대한 비용을 최소화하기 위해 동적 이동 경로 생성 또는 미리 점검하는 기능을 제공한다.

에코전트가 이동이 결정되면 각각의 에코전트들은 현재의 플랫폼 및 네트워크의 상태를 통해 이동 경로가 생성된다. 생성된 이동 경로에 따라 에코전트의 수행 시간 및 코드의 이동에 대한 네트워크 부하를 최소화하기 위한 선행 작업으로, 목적지 플랫폼에서 운영될 코드를 미리 수행시키기 위한 프리패칭(prefectching) 기능을 수행한다. 각 플랫폼에서는 프리패칭 요청에 따라 에코전트의 수행에 필요한 코드의 존재여부를 판단하고 필요한 코드가 존재하는 플랫폼에 에코전트의 코드 전송을 요청하고 이를 수신하여 플랫폼에 기록하는 기능을 수행한다. 특정 플랫폼에서 에코전트의 작업이 완료되면 각 플랫폼에 존재하는 에코전트들은 이동 경로에 따라 다른 플랫폼으로 이동하여 필요한 작업을 수행하게 된다.

그러나 에코전트가 특정 플랫폼에서 작업을 수행하는 과정에서 이동할 플랫폼 또는 네트워크의 상태에 따라 에코전트가 실제적으로 이동할 수 없거나 또는 이동한 플랫폼에서 에코전트를 수행시키지 못할 경우가 발생한다. 따라서 에코전트가 특정 플랫폼으로 이동할 코드 및 데이터를 전송하기 전에 에코전트가 해당 플랫폼으로 이동 가능한지를 판별하고 이동이 불가능 할 경우 이동할 경로를 동적으로 생성한다. 동적인 이동 경로의 생성은 플랫폼의 상태 또는 네트워크의 상태에 따라 최소의 비용으로 이동 가능한 경로를 생성한다.

서바이버빌리티서비스는 플랫폼의 에러 또는 에코전트 자체의 문제로 인해 해당 작업을 수행하지 못하고 중지되는 상황을 막고자 제안된 서비스이다. 에코전트의 폴트 톨러런트(fault-tolerant)한 기능을 수행하기 위해서는 에코전트의 상태를 주기적으로 기록하거나 에코전트의 복사본을 동시 수행시키는 기법들을 수행할 수 있다.

에코전트의 폴트 톨러런트한 수행을 위해 NRAMA (N-Replication And M-Activation) 정책과 체크 포인트 기능을 제공한다. NRAMA는 에코전트의 중지 시간을 최소화

하기 위해 N개의 에코전트의 복사본을 유지하고 필요에 따라 M개의 복사본을 동시에 수행시키는 기능이다. 만약 N이 0이라면 에코전트의 복사본이 수행되지 않기 때문에 이러한 상황에서는 체크 포인트 기능을 이용하여 에코전트의 상태를 주기적으로 기록할 수 있다. 또한 N이 1이상이라면 현재 수행 중인 에코전트와 동일한 기능을 수행하는 에코전트를 생성하고 그 중 M개를 동시에 수행시키는 기능을 제공한다. 동시에 수행되는 에코전트의 개수 M이 많을수록 플랫폼 또는 에코전트의 에러에 의해 중지되는 시간은 최소화될 수 있다. 그러나 동시에 동일한 기능을 다수의 에코전트가 수행하기 때문에 시스템의 성능을 저하시킬 수 있다. 또한 동시에 수행되는 에코전트 중 하나의 에코전트만을 완료시키기 원할 경우 완료 정책이 복잡해질 수 있다. 에코전트의 복사본의 수행과 함께 에코전트의 상태를 기록하기 위한 체크 포인트 기능을 제공한다.

체크 포인트는 에코전트의 상태를 에코전트 개발자에 의해 안전한 저장 장치에 기록할 수 있다. 일반적으로 체크 포인트는 에코전트 개발자에 의해 선택적으로 수행될 수 있다. 그러나 특정 플랫폼에서 에코전트가 처음 수행되거나 에코전트가 이동하기 전에는 항상 체크 포인트 기능을 수행하여 활동 중이거나 이동하는 에코전트를 상실하는 문제점을 제거할 수 있다. 에코전트 또는 플랫폼에 에러가 발생하면 에코전트의 NRAMA 정책과 체크 포인트 기능을 통해 에코전트의 상태를 복구하거나 중지시키지 않고 에코전트를 계속적으로 수행할 수 있다. 만약 에코전트가 NRAMA 정책을 사용하여 복사본을 수행하고 있었다면 에코전트의 우선순위에 따라 에러가 발생한 에코전트의 복사본이 수행 중인지 확인한다. 만약 에코전트의 복사본이 수행 중이라면 에러가 발생한 에코전트는 활성화되지 않는다. 그러나 에코전트의 복사본에 대한 수행이 없거나 체크 포인트 기능만을 수행하였다면 체크 포인트를 통하여 저장된 에코전트의 상태를 확인하고 체크 포인트 이후의 작업을 계속 수행한다.

3.2.2 컨테이너

컨테이너는 로컬 리소스 매니지먼트, 에코전트 모니터링, 에코전트 스케줄링, 에코전트 레지스트레이션의 5가지 컴포넌트로 구성되어 있다. 이는 시스템에 접근하여 에코전트 플랫폼 레이어의 물리적인 기능을 담당해 주는 부분이다.

로컬 리소스 매니지먼트 컴포넌트는 하나의 에코전트 플랫폼의 모든 자원을 모니터링하고 관리 하는 기능을 한다. 여기서 제시 하는 모델은 JVM(java virtual machine)을 기반으로 하고 있다. 따라서 JVMPI[4]를 이용하면 이러한 구현이 가능하다. 이러한 모니터링기능을 이용하여 에너지 매니지먼트 서비스가 해당 에코전트에 대한 자원 사용량을 질의하면 알려주게 된다. 또한 이를 활용하여 자원의 한계치를 적절히 조절하여 에코전트의 활동을 유지 시켜 주게 된다. 마지막으로 다른 플랫폼으로 이동하고자 할 때에 리모트 플랫폼에 새로운 에코전트가 활동가능한지 여부도 알려 줄 수 있다.

에코전트 모니터링 컴포넌트는 에코전트에 대한 상태 정보 및 메타 데이터들을 관리하고 제공하는 역할을 한다. 에코전트 컴포넌트의 역할은 크게 두 가지로 나뉜다. 첫째는 에코전트에 대한 정보를 제공하는 것이고 두 번째는 에코전트의 상태를 모니터링하여 변화가 발생하면 이를 알려 주는 기능을 하게 된다.

에코전트 스케줄링 컴포넌트는 플랫폼 상에서 동작하고 있는 에코전트를 스케줄링하는 역할을 한다. 에코전트 쓰레드의 우선순위(priority)를 조정하고, 에코전트를 멈추게 하거나(suspend) 다시 실행 시키는(wakeup) 역할을 한다. 또한 이러한 결과는 에코전트 레지스트레이션 컴포넌트에 등록하게 된다.

에코전트 레지스트레이션 컴포넌트는 로컬 플랫폼 상에 동작하고 있는 에코전트에 대한 정보를 저장하는 역할을 한다. 에코전트의 GUID(Global Unique ID)를 바탕으로 구

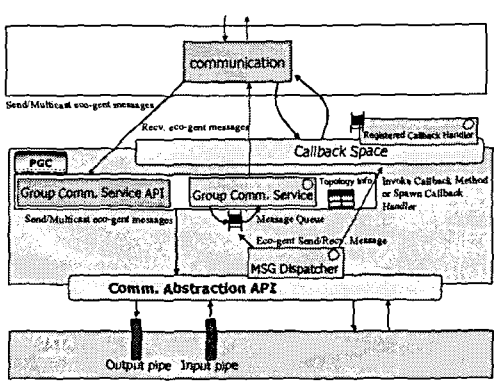
성된 에코전트의 상태 정보들을 관리하고 저장하게 된다.

3.3 플랫폼 콜레버레이션 레이어

플랫폼 콜레버레이션 레이어는 2개의 서브 레이어로 구성되어있다. ([그림 1] 참조) 피어-투-피어 커뮤니케이션 서브스트레이트(Peer-to-Peer Communication Substrate, PCS)는 피어-투-피어 환경에서 에코전트 플랫폼이 수행되기 위한 기본적인 서비스를 제공하고 있다. JXTA[6]는 인터넷상에 가상의 피어-투-피어 네트워크를 형성하여, 네트워크 구조에 상관없이 피어들 간에 상호 작용이 가능하게 해준다.

3.3.1 플랫폼 그룹 커뮤니케이션 서비스

플랫폼 그룹 커뮤니케이션 서비스(Platform Group Communication Service, PGC)는 피어-투-피어 환경에서 다른 플랫폼들과 협동 작업과 플랫폼 그룹 통신 서비스를 제공한다. PGC는 [그림 3]에 나타나있는 것처럼 에코전트 플랫폼 레이어의 커뮤니케이션 서비스에 통신 API를 제공하고, 에코전트의 send/recv. 메시지를 처리함으로써 동기적 통신을 지원한다. 또한, 콜백 공간에 콜백 함수를 등록함으로써 큐를 거치지 않고도 메시지를 바로 전달하는 비동기적 통신을 지원한다.

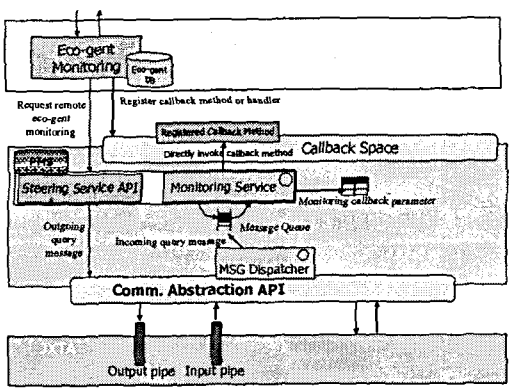


[그림 3] 플랫폼 그룹 커뮤니케이션 서비스

3.3.2 플랫폼 모니터링 서비스

플랫폼 모니터링/스티어링 서비스(Platform Monitoring and Steering service, PMS)는 에코전트 플랫폼간의 쿼리 메시지를 처리하고, 플랫폼의 콜백 파라미터를 조정하는 역할을 한다. ([그림 4] 참조)

예를 들어, 에코전트 모니터링 서비스는 모니터링 하고자 하는 에코전트의 리스트를 PMS의 스트어링 서비스 API를 통해 전달하고, 해당 플랫폼은 모니터링 서비스 콜백 함수의 파라미터로 이를 설정한다. PRC의 컨트롤 서비스는 이 파라미터를 가져와 주기적으로 콜백 함수를 호출하게 된다. PMS의 모니터링 서비스는 메시지의 종류에 따라서 직접 콜백 함수를 호출할 수 있다.



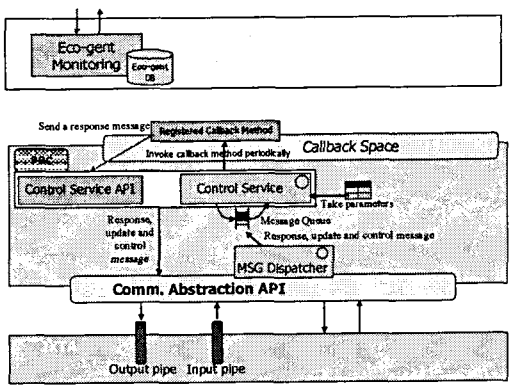
[그림 4] 플랫폼 모니터링/스티어링 서비스

3.3.3 플랫폼 런타임 서비스

플랫폼 런타임 컨트롤 서비스(Platform Runtime Control service, PRC)는 플랫폼의 가입과 탈퇴를 플랫폼 그룹에 알리고, 플랫폼의 상태가 바뀌면 업데이트 메시지를 내보낸다. ([그림 5] 참조)

PRC의 컨트롤 서비스 API를 사용해서 콜백 공간의 콜백 함수들은 호출한 플랫폼에 대해 응답 메시지를 보낼 수 있다. 예를 들어, 앞에서 언급했던, 모니터링 서비스 콜백의 파라미터를 통해서 모니터링 서비스를 콜백을

호출하면, 콜백 함수 내부에서는 컨트롤 서비스 API를 통해 응답 메시지를 받고자하는 플랫폼에 메시지를 전달한다. 이때 콜백을 주기적으로 호출할 수 있는데, 이는 파라미터의 값으로 명시되어야 한다.



[그림 5] 플랫폼 런타임 컨트롤 서비스

4. 응용 분야에 대한 고찰

생태계 모방 시스템 소프트웨어를 기반으로한 에코전트의 응용은 기존의 분산 시스템이나 멀티 에이전트가 담당했었던 분야에 대한 적용과 바이오 기능을 필요로 하는 새로운 분야에 대한 적용으로 나누어 생각해 볼 수 있다.

분산 웹 서버는 멀티 에이전트가 사용되었던 분야에 대한 좋은 예이다. 분산되어 있는 각 웹 서버를 에코전트로 구성하였다고 가정하여 보자. 특정한 시간대나 지역에 클라이언트의 요청이 몰린다면 웹 서버-에코전트의 복제나 이동을 통해서 원할 한 웹 서비스를 제공할 수 있다. 또한 클라이언트의 요청이 많지 않은 지역이나 시간대에는 웹 서버의 수를 줄여서 자원의 낭비를 막을 수도 있을 것이다. 일정 지역에 대한 최소 개체 생존 보장이나 개체간의 협력 기능은 견고한 분산 웹 서버를 구성하는데 도움을 줄 수 있다. 웹 서버인 Aphid[5]는 바이오 에이전트로서 디자인되었으며, 실험을 통해 일반적인 분산 웹 서버보다 좋은 성능을 보여줄 수 있음이 확인되었다. 이렇듯 기존의 분산 시스템이나 멀티 에이전트를 사용하여 구성된 시스템에서 에코전트를 사용함으로써 시스템의 적응성과 견고성을 향상 시킬 수 있다.

에코전트는 기존의 문제 해결뿐만 아니라 새로운 목적을 가진 시스템을 구성하는 데에도 효과적으로 사용될 수 있을 것이다. 예를 들어, 도로와 차들을 에코전트로 구성하여 도시에 맞는 최적의 도로 조건을 스스로 찾아내는 시스템을 구성할 수도 있을 것이다. 이러한 시스템은 도로의 상황에 적절한 수와 활동을 가능하게 하는 에코전트의 발생을 유도할 수 있다. 또한 에이전트화된 백신을 통해 좀 더 지능적인 바이러스 퇴치 서비스를 제공할 수도 있을 것이다.

5. Conclusion

미래의 네트워크 환경에 적응하기 위하여 생태계의 구조를 모방한 (Bio-inspired) 네트워크 기반 시스템을 제안하였다. 이는 적응성(adaptability), 확장성(scalability), 생존성(survivability)이라는 생태적 특징들을 기반으로 하였다. 이러한 모방이 생태계의 조화로운 구조를 모두 수용할 수 있다고 보지는 않는다. 하지만 생태계의 특징들을 적절히 분석하고 이용하는 것이 급속히 변하는 네트워크 컴퓨팅 환경에 대응하는 지름길이라고 생각한다.

References

[1] Stigmergic systems, http://www.stigmergicsystems.com, 2003

[2] Tatsuya Suda. Adaptive Networking Architecture for Service Emergence. 2001 Symposium on Applications and the Internet-Workshops (SAINT 2001 Workshops) January 08 - 12, 2001

[3] O.Rana and K.Stout. What is Scalability in Multi-Agent System. Proceedings Autonomous

Agents 2000, Barcelona, pages 56-63, 2000.

[4] JVMPI (JavaTM Virtual Machine Profiler Interface), http://java.sun.com/j2se/1.4.2/docs/guide/jvmpi/jvmpi.html

[5] Wang.M. and Suda.T., "The Bio-Networking Architecture: A Biologically Inspired Approach to The Design of Scalable, Adaptive, and Survivable/Available Network Applications," *the proceedings of applications and the Internet 2001*, pp.43-53, 2001.

[6] Bernard Traversat, Ahkil Arora, Mohamed Abdelaziz, Mikr Duigou, Carl Haywood, Jean-Christophe Hugly, Eric Pouyoul and Bill Yeager Project, JXTA 2.0 Super-Peer Virtual Network, http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf