

:

KLDP

1), ,

가

가 ,

KLDP(Korea

Linux Documentation Project)

KLDP

가 ,

가

가 가

Key Words :**1.**

정보기술의 발전과 인터넷의 보급은 사회 시스템의 구조와 구성원 간에 일어나는 상호작용 방식에 근본적인 변화를 가져왔다. 기존의 학연, 혈연, 지연 등 공간적, 배경적 제약이 강한 유대 관계 중심에서 벗어나 취미 및 여가활동, 그리고 정보 교환 등을 위해 시공간의 제약이 없는 인터넷 상에서 커뮤니티와 거래의 장터가 만들어지고 있다. 이는 인터넷이 정보의 탐색비용(search costs)과 인적 물적 네트워크를 구성하는데 들어가는 거래 및 조정비용(transaction and coordination costs)을 낮추었기 때문이다.

특히, 사회 경제적 관점에서 인터넷의 가장 큰 역할은 최신의 가치 있는 정보를 빠른 속도로 확산시킨다는 점이다. 뉴스를 실시간으로 검색하고, 인터넷 포털 사이트의 지식 검색 서비스를 이용하여 간단한 정보의 질의 응답이 이루어지며, 위키피디아(WIKIPEDIA), 각종 블로그(blog) 등을 통해 고급 정보를 구하는 지식 공유 활동은 지식의 공공재(public

1)E-mail : wych@kgs.m.kaist.ac.kr

goods)적 특성을 고려해 볼 때, 사회적으로 바람직한 현상임을 알 수 있다.

그런데, 최근에 인터넷은 단순히 지식 정보의 공유 및 확산을 위한 유통망(diffusion network)으로서의 역할에서 벗어나 새로운 정보재화(information goods)를 생산하는 가치 창출의 수단으로 활용되고 있다. 가장 대표적인 사례가 컴퓨터 프로그램 개발자들이 자발적으로 참여하여 운영되는 공개소프트웨어(open source software) 개발 커뮤니티이다. 공개소프트웨어란 프로그램의 소스코드가 공개되어 있어 누구나 자신의 필요에 맞게 프로그램을 수정 및 개선할 수 있고 이를 다른 사람들에게 제공하는 것이 자유롭지만 공개소프트웨어 라이선스에 의해 상업적 판매가 금지된 소프트웨어를 의미한다.¹⁾ 상용소프트웨어(proprietary software)는 소스코드가 공개되지 않기 때문에 프로그램의 오류가 발생하거나 추가적인 기능에 대한 업그레이드가 필요할 경우 개발업체에게 의존해야 하는 불편함이 있는데, 이러한 문제를 해결하려는 요구가 공개소프트웨어의 개발로 이어지는 것이다. 실제로 이러한 요구는 개발자들이 사용하는 서버 수준의 시스템에서 상대적으로 빈번하게 발생하기 때문에 전문화된 시스템 환경에서 공개소프트웨어가 많이 활용된다. 예를 들어 2005년 현재 전세계 웹 서버시장은 공개소프트웨어인 아파치가 70%에 육박하는 시장 점유율로 상용소프트웨어인 마이크로소프트의 IIS를 세 배 이상 앞서고 있다.

대부분의 공개소프트웨어는 상업적 목적이 없는 개발자들의 자발적인 인터넷 커뮤니티 활동을 통해서 개발된다. 즉, 특정 소프트웨어 개발을 추진하는 프로젝트 운영자가 개발하고자 하는 소프트웨어에 대한 소개와 개발 방법, 버전 출시 계획을 포함한 프로젝트 운영 계획, 그리고 인력 모집 등에 대한 정보를 공개소프트웨어 커뮤니티 사이트에 공지하면 이에 관심 있는 개발자가 프로젝트에 지원하고, 지원자 중 일정한 절차를 통해 선발된 개발자들의 활동으로 소프트웨어 개발이 이루어진다.

공개소프트웨어를 통해서만 경제적인 이윤을 얻을 수 없을 뿐만 아니라 수평적으로 분화되어 통제와 조정이 어려운 인터넷 기반 네트워크 구조에서 개발이 이루어지기 때문에 공개소프트웨어의 개발 및 사용에 한계가 있을 것이라는 일반적 예상에도 불구하고 공개소프트웨어를 개발하거나 이를 사용하는 인구는 점차 늘고 있다. 전세계에서 가장 큰 공개소프트웨어 개발 커뮤니티인 SourceForge사이트²⁾의 경우, 2006년 7월 현재 12만개 이상의 소프트웨어 개발 프로젝트가 운영되고 130만 이상의 사용자들이 등록되어 2001년에 비해 5배 이상의 성장을 기록하였다. 국내의 대표적인 공개소프트웨어 커뮤니티인 KLDP(Korean Linux Documentation Project)³⁾는 2006년 7월 500여 개에 달하는 프로젝트와 1만 이상의 사용자가 등록되어 있다.

다른 인터넷 커뮤니티와 마찬가지로 공개소프트웨어 커뮤니티 활동 역시 분산된 네트워크에서 다양한 동기를 가진 자발적 참여자들의 복잡한 상호작용을 통해 이루어진다. 또한 커뮤니티 안에는 지속적으로 새로운 개발 프로젝트가 생성되고 성과가 높지 않은 프로젝트는 중간에 사라지며 프로젝트를 성공적으로 수행한 개발자들은 명성이 높아지고 프로그래밍 능력도 향상되어 이들이 참여하는 프로젝트에는 개발자들이 많이 모이는 양의 되먹임 현상이 나타나는 등 공개소프트웨어 커뮤니티는 복잡하고 동태적인 특성을 지니고 있다. 이러한 공개소프트웨어 커뮤니티의 특성을 정리하면 다음과 같다.

1) Open Source Initiative (www.opensource.org)
가

2) www.sourceforge.net

3) www.kldp.net

- 평평하고 분산화 된 네트워크 구조임(flat and decentralized network)
- 참여자들은 이질적 특성과 다양한 동기를 가지고 있음(heterogeneous agents)
- 참여자들이 자발적으로 프로젝트를 조직하여 활동함(self-organized system)
- 시간이 지나면서 네트워크 형태와 개발자 특성이 변화함(dynamic system)
- 특정 시간이나 참여자 수준을 넘어서면 새로운 질서가 관찰됨(emergence)
- 참여자간 조정과 협업 등 복잡한 상호작용이 이루어짐(non-linear interaction)
- 지속적으로 새로운 참여자가 외부에서 유입됨(open system)

이러한 특성 때문에 공개소프트웨어 커뮤니티는 수많은 이질적인 구성요소들이 비선형적 상호작용을 통해 얽혀 있고 거시적으로 보았을 때 새로운 질서나 급격한 변화가 자주 목격되는 복잡적응시스템(complex adaptive system)¹⁾, 혹은 자기조직화 시스템(self-organizing system)으로서의 특성을 가지고 있다고 볼 수 있다.

따라서 공개소프트웨어 커뮤니티에 대한 조직론적 연구, 개발자들의 참여 동기와 성과에 대한 연구, 개발자간의 협업 메커니즘 연구 등은 복잡계 이론의 틀 안에서 이루어질 필요가 있다. 실제로 최근에는 공개소프트웨어의 개발과 확산 현상을 복잡계 시스템 차원에서 분석하는 연구가 점차 늘어나고 있다. Madey et al. (2002)는 SourceForge 사이트의 데이터를 활용하여 행위자 기반 모형을 구축하는 연구를 수행하였고, Khalak (2003)은 소프트웨어 시장에서 공개소프트웨어와 상용소프트웨어간의 경쟁상황을 행위자 기반 모형을 활용하여 분석하였다. 또한 Navarro (2004)는 중남미 지역에서 인터넷의 보급이 확대되면서 관찰되는 인터넷 기반 자기조직화 네트워크의 사례로 공개소프트웨어를 소개한 바 있으며, Xu et al. (2005)은 SourceForge를 대상으로 네트워크 위상구조를 분석하였다. 한편 Valverde et al. (2006)은 공개소프트웨어 커뮤니티와 말벌(wasp) 군집체의 네트워크 위상구조적 특성을 비교 연구하였다. 언급한 대부분의 연구가 탐색적 연구의 성격이 강하지만 학계에서도 공개소프트웨어 현상을 올바르게 이해하기 위해 새로운 분석도구가 필요하다는 합의가 이루어지는 추세임을 알 수 있다.

2.

본 연구는 복잡계 연구 방법론을 이용하여 국내 공개소프트웨어 개발 커뮤니티인 KLDP 사이트²⁾를 대상으로 네트워크 위상구조(network topology)적 특성을 파악하는 것을 목적으로 하고 있다. 본 연구의 결과는 향후 커뮤니티의 안정성과 동적 성장 방향, 그리고 커뮤니티

1) (2006)
 2) KLDP 1996 10, geekforum, 2002
 2000 11, kldp.net, KLDP
 7, BBS, wiki가, www.kldp.org
 www.kldp.net, www.kldp.net

티 활성화를 위한 정부의 정책수립을 위한 정책적 제언에 기초자료로 활용될 수 있다는 점에서 의의가 있다고 판단된다.

본 연구는 다음의 연구 문제에 대한 답을 도출하는 것을 목표로 하고 있다.

■ KLDP 커뮤니티의 네트워크 위상구조적 특성 파악

소수의 참여 개발자들의 역할이 중요한가? 역할이 고르게 분포되어 있는가?
참여자들이 얼마나 긴밀하게 연결되어 조직화되었는가? 좁은세상효과(small world phenomenon)나 척도 없는 네트워크(scale-free network)의 특성을 관찰할 수 있는가?

결점 간에 유유상종 현상이 나타나는가? 즉 위상구조적으로 중요한 역할을 하는 결점들끼리 서로 연결되어 있는가?

■ 네트워크의 위상구조적 속성과 개별 결점 속성 간의 관계 파악

네트워크 상에서 근접해있거나 군집으로 묶이는 결점 간에 어떠한 개별적 속성을 공유하고 있는가?

■ 다른 사회 네트워크와의 비교 분석

SourceForge와 같은 대규모 공개소프트웨어 커뮤니티와 비교할 때 유사점과 차이점은 무엇인가?

연구 방법 및 절차는 다음과 같다. 우선, KLDP 사이트에서 프로젝트 및 개발자와 관련된 자료를 수집하여 적절한 자료 형태로 변환한다. 그 다음으로 네트워크 위상구조 분석 소프트웨어인 넷마이너(NetMinder)를 활용하여 공개소프트웨어 커뮤니티의 위상구조적 특성을 파악한다. 마지막으로 분석한 결과와 기존의 공개소프트웨어에 대한 연구 결과를 활용하여 위에서 언급한 연구 문제에 대한 해석을 도출한다.

복잡계 관점에서 공개소프트웨어 커뮤니티를 분석하는 것은 앞서 언급한 바와 같이 공개소프트웨어의 개발 및 확산과정이 복잡계 시스템의 전형적인 모습을 띄고 있기 때문이며 이를 통하여 연구대상의 특성을 올바르게 이해할 수 있을 뿐만 아니라 향후 커뮤니티의 동태적 변화과정을 예상해 볼 수 있다. 이와 함께 본 연구는 국내의 사례 연구로서 국내 복잡계 연구의 논의를 더욱 풍부하게 하는데 기여할 수 있으며, 향후 커뮤니티 참여유인에 대한 설문 및 행위자 기반 모형(agent-based model)을 구축하는 등의 공개소프트웨어 커뮤니티 연구에 필요한 기반을 제공한다는 점에서 특히 의미가 있을 것이다.

3.

공개소프트웨어 커뮤니티에서는 다수의 프로젝트가 운영이 된다. 각각의 프로젝트는 상이한 종류의 소프트웨어를 개발하는 것을 목표로 하고 있으며, 그 목표를 달성하기 위해 필요한 개발자의 참여를 유도한다. 개발자의 입장에서는 자신에게 필요한 소프트웨어를 개발하고자 하는 욕구가 있다. 이 때 개발자는 자신에게 특화되어 있거나 전문성이 높은 개발언어를 사용하기 원하며 사용하는 운영체제와 환경을 고려하여 프로젝트가 진행되기를 원하게 된다. 이러한 욕구를 충족시키기 위해서 개발자는 기존의 프로젝트에 참여하거나 새로운 프로젝트를 개설하여 이해를 같이 하는 개발자의 참여를 유인하게 된다. 따라서

공개소프트웨어 커뮤니티는 다수의 프로젝트와 자신에게 적합한 프로젝트에 선택적으로 참여하는 개발자들로 구성된 공동참여 네트워크(affiliation network)라고 이해 할 수 있다. 이때 공개소프트웨어 커뮤니티 네트워크는 개발자를 결점으로 구성한 개발자 네트워크와 프로젝트를 결점으로 구성한 프로젝트 네트워크 등 두 종류의 네트워크를 구성할 수 있게 된다.

간단히 예를 들면, Figure 1은 6개의 프로젝트에 11명의 개발자가 참여하고 있는 공동참여 네트워크를 구성한 것이다. 좌측의 개발자 네트워크에서는 개발자를 결점으로 삼고, 공동으로 참여하는 프로젝트가 있는 개발자끼리 연결하여 네트워크를 구성하였고, 우측의 프로젝트 네트워크의 경우 프로젝트를 결점으로 삼아 프로젝트 간에 공유하는 개발자가 있는 경우 연결시킴으로써 네트워크를 구성하였다.

이 때, 개발자 네트워크와 프로젝트 네트워크의 경우 공통적으로 C1과 C2라는 두 개의 군집(cluster)으로 분리되는 것을 확인할 수 있는데, 이는 개발자끼리 공동으로 참여하는 프로젝트가 전혀 없어 두 그룹간에 연결고리를 찾을 수 없는 경우이다. 네트워크가 이러한 군집으로 나뉘어 있을수록 의사소통과 전체적인 조율에 어려움이 많아지게 된다.

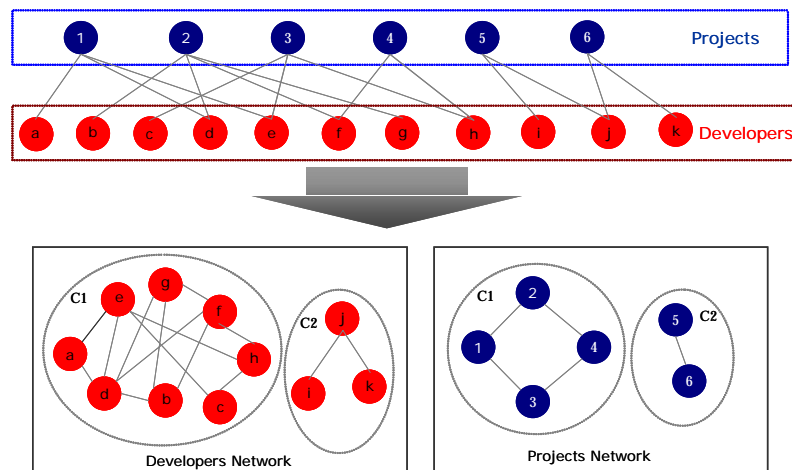


Fig. 1 Formations of Developers/Projects Network in Affiliation Network

본 연구는 국내 공개소프트웨어 개발 커뮤니티 중 KLDP 사이트 (www.kldp.net)에 2006년 10월 1일 현재 공개된 프로젝트 429개와 이에 참여하는 590명의 개발자 데이터를 수집하여 네트워크 위상구조를 파악하였다. 본 절에서는 KLDP 사이트의 위상구조를 크게 개발자 네트워크와 프로젝트 네트워크로 나누어 각각 기본적인 구조적 특성을 파악하고 이를 해석하고자 한다. Figure 2은 KLDP 커뮤니티의 개발자 네트워크를 결점과 링크로 표현한 그림이다.

3.1

공개소프트웨어 커뮤니티 네트워크에서 개발자는 프로젝트에 참여함으로써 공동으로 참여한 개발자들과 연결관계를 갖게 된다. 따라서 이러한 네트워크에서 개발자가 다수의 프로젝트에 참여할수록, 그리고 프로젝트의 규모(하나의 프로젝트에 참여하는 개발자의 수)가 커질수록 다수의 개발자와 연결관계를 맺게 되어 네트워크의 연결수준이 높아진다. 공개소프트웨어 커뮤니티의 경우 전체 개발자의 82%에 해당하는 483명의 개발자가 한 개의 프로젝트에만 참여하고 있으며 일인당 평균 1.35개의 프로젝트에 참여하고 있다. 또한 전체 프로

젝트의 65% 수준인 277개의 프로젝트가 한 명의 개발자에 의해서 운영되고 있으며 프로젝트 당 평균 1.86명의 개발자가 참여하고 있다.

이를 통해 공개소프트웨어 커뮤니티가 매우 느슨하게 조직되어 연결수준이 낮은 네트워크라는 것을 알 수 있다. 개발자 일인당 참여하는 프로젝트의 숫자가 적은 이유는 공개소프트웨어를 통해 직접적인 경제적 이윤을 얻기 힘들고 대부분의 개발자들이 개인의 필요에 의해서 자발적인 여가 및 취미활동으로서 참여한다는 점과 이러한 개인적 필요는 대중적으로 요구되지 않는 경우가 많다는 점에서 그 원인을 찾을 수 있다. 소프트웨어 개발에는 일정수준의 개발 능력과 투입 시간이 요구되는데 경제적 유인이 부족한 상황에서 개발자들이 투입할 수 있는 시간과 노력에는 한계가 있다. 한편 프로젝트의 규모가 작은 이유는 국내의 공개소프트웨어 인식수준이 낮거나 개발할만한 소프트웨어가 많지 않거나 혹은 개발되는 소프트웨어의 적절 개발 인원이 적기 때문으로 추론해볼 수도 있다.

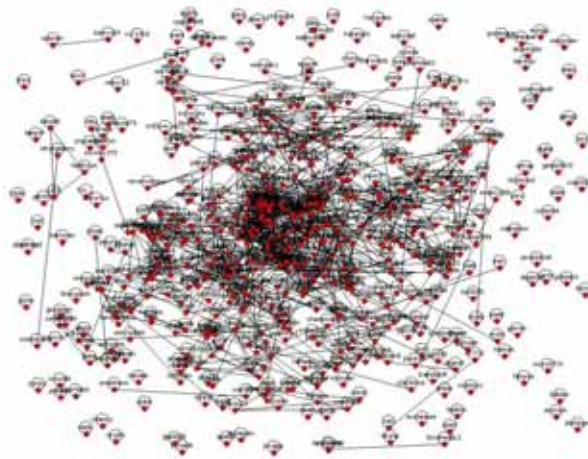


Fig. 2 KLDP Developers Network

실제로 KLDP 네트워크에 참여하는 개발자가 몇 명의 다른 개발자와 연결되어 있는지에 대한 측정 지표인 평균 연결수(average degree)는 3.69임을 알 수 있었다. 이는 개발자가 평균적으로 3.69명의 다른 개발자들과 연결관계를 맺고 있음을 의미한다. 또한 네트워크 참여 개발자 중 약 31%에 해당하는 181명의 개발자는 다른 개발자와 전혀 연결관계를 맺지 못한 채 소프트웨어를 개발하는 것으로 나타났다. 물론 공개소프트웨어 커뮤니티를 개발자뿐만 아니라 사용자까지 포함시키는 네트워크로 구성한다면 이들이 개발한 소프트웨어를 다른 참여자들이 사용하거나 이들이 다른 공개소프트웨어를 사용함으로써 네트워크에 편입될 수도 있겠지만, 개발자만으로 구성된 네트워크에서는 동떨어진 결점으로서 존재하게 된다.

개발자들이 보유한 연결수의 상대적 빈도의 분포함수를 연결수 분포(degree distribution)라고 한다. Barabasi and Albert (1999)는 네트워크가 성장하는 과정에서 새롭게 네트워크에 진입하는 결점은 기존의 결점 중에서 이미 연결수가 많은 결점과 연결관계를 맺으려는 성향이 있기 때문에 연결수에 있어서 빈익빈-부익부 현상이 나타나며 이를 척도 없는 네트워크(scale-free network)라고 명명하였다. 이러한 특성을 가지고 있는 네트워크의 연결수 분포를 로그-로그(log-log)함수로 전환하여 그려보면 우하향하는 직선의 형태를 보이게 되며 이 분포가 거듭제곱법칙(power law)이 성립함을 의미하게 된다.

공개소프트웨어 개발자의 중요한 참여유인으로 명성을 얻으려는 동기가 있는데 이는 자신의 능력을 다른 개발자들에게 보임으로써 자존감(self-esteem)을 높이거나(Weber, 2004) 자신의 능력을 보임으로써(signaling effect) 경력을 개발하여 향후 좋은 직장을 구하는데 도움을 받기 위해 프로젝트에 참여하는 동기(Lerner and Tirole, 2002)를 일컫는다. 일반적으로 명성을 얻으려는 동기를 가진 개발자는 커뮤니티 내에서 다수의 다른 개발자들과 관계를 형성하고 이들과의 협업을 통해서 자신의 능력과 명성을 보이기 위해 노력하는데 실제로 개발자들이 인적 네트워크를 쌓기 위해 다수의 프로젝트에 참여하거나 프로젝트 중에서 다수의 개발자가 이미 참여하고 있는 프로젝트에 참여하는 것을 선호한다.

하지만 위의 첫 번째 활동은 개발자의 개발 능력에 한계가 있기 때문에 무수히 많은 프로젝트에 참여하는 것이 불가능하다는 점이 있고, 두 번째 활동은 개발자 수가 많아지면 오히려 의견 조율과 협업이 어려워질 수 있기 때문에 프로젝트 운영자가 많은 수의 개발자의 참여를 허락하지 않을 가능성이 높다. 따라서 공개소프트웨어 커뮤니티에서는 멱함수 법칙이 성립하지 않을 수 있다.

그럼에도 불구하고 Figure. 3은 공개소프트웨어 커뮤니티에서 멱함수의 법칙이 성립함을 보여준다. Figure. 3의 (a)는 프로젝트 규모, 즉 개별 프로젝트에 참여하는 개발자 수의 분포함수를 로그-로그(log-log)함수로 전환한 결과이고 (b)는 개발자 1인이 참여하는 프로젝트의 분포함수를 역시 로그-로그(log-log)함수로 전환한 결과이다. 두 함수 모두 선형회귀분석의 R^2 값이 0.9 이상으로 우하향하는 직선에 상당히 근사 하는 것을 확인할 수 있다.

개발자 네트워크의 연결수준을 파악하는 또 다른 측정도구로는 임의의 두 결점 간에 몇 단계를 거치면 서로 연결이 되는지를 알려주는 평균 거리(mean distance)가 있다. 분석결과 본 공개소프트웨어 네트워크의 평균 거리는 3.3 단계인 것으로 나타났다. 결점 수가 45만 개 정도되는 영화배우 네트워크의 평균 거리가 3.5였고(윤영수, 채승병, 2006), Xu et al. (2005)에 의하면 또 다른 공개소프트웨어 커뮤니티인 SourceForge의 경우 2003년 현재 83,118명의 개발자를 대상으로하여 평균 거리가 10.2인 것으로 분석되었다. 이러한 사실들에서 현재 국내의 KLDP 사이트의 개발자 네트워크는 아직 작은 규모이기 때문에 영화배우 네트워크와 비슷한 수치가 나왔다고 판단할 수 있을 것이다. 그리고 공개소프트웨어 네트워크는 평균적으로 볼 때 다른 네트워크보다 평균거리가 더 클 것으로 예상해 볼 수 있고 이것은 공개소프트웨어의 공공재적 특성에 기인한다고 판단된다.

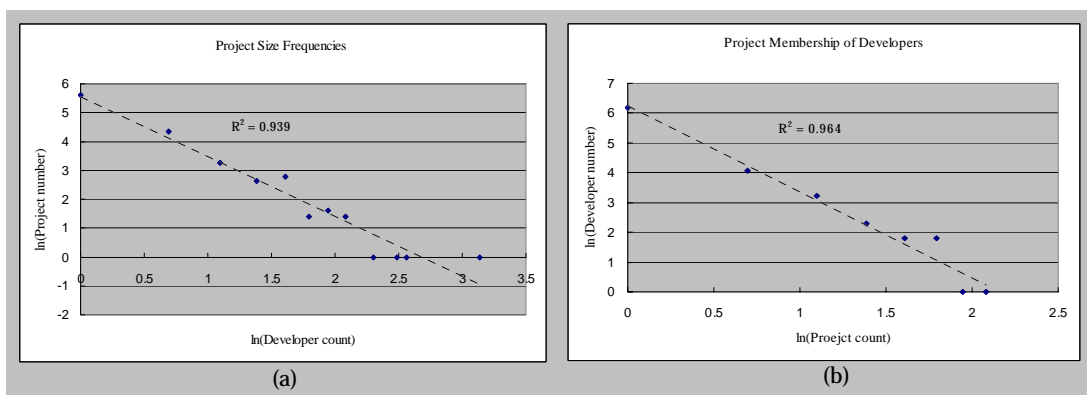


Fig. 3 Log-log Transformation of (a) Project Size Frequencies and (b) Project Membership Developers

3.2

네트워크를 구성하는 결점 간에 얼마나 긴밀하게 연결되어 있는지를 살펴보는 측정도구는 밀도(density)와 결속계수(clustering coefficient)가 있다. 밀도는 네트워크의 실제 연결수를 네트워크의 모든 결점이 연결되어 있을 경우 가능한 최대의 연결수로 나눈 값으로서 네트워크 전체의 연결의 긴밀한 정도를 표현한다. 결속계수는 특정 결점과 연결된 결점 간에 얼마나 긴밀하게 연결되어 있는지를 계산하는 측정도구이다. 측정 결과 KLDAP 커뮤니티는 밀도가 0.006으로서 상당히 낮은 수준이지만 결속계수는 0.953으로 매우 높게 나타났다. 일반적으로 밀도가 높으면 결속계수가 높아지게 된다. 하지만 결속계수가 높다는 것이 밀도가 높다는 것을 의미하지는 않음을 보여준다.

이러한 특성은 KLDAP 커뮤니티 네트워크가 공동참여 네트워크라는 점을 고려해볼 때 다음과 같이 설명될 수 있다. 우선 밀도가 매우 낮은 이유는 앞서 설명한 바와 같이 공개소프트웨어 개발 커뮤니티 프로젝트에서 요구하는 개발자 규모와 개발자들이 참여할 수 있는 프로젝트의 숫자에 한계가 있다는 점에서 설명이 가능하다. 하지만 특정 프로젝트에 참여하는 참여자는 그 프로젝트내의 모든 참여자와 연결된다는 공동참여 네트워크의 특성을 고려해보면 결속계수는 매우 높게 나오게 되어 밀도는 낮고 결속계수는 높게 나타나는 현상이 설명된다.

3.3

네트워크에서 차지하는 결점의 상대적 중요도는 중앙성 지표를 통해서 파악할 수 있다. 일반적으로 연결 정도 중앙성(degree centrality), 사이 중앙성(betweenness centrality), 그리고 위세 중앙성(Eigenvector centrality) 등이 많이 활용되고 있다. 연결정도 중앙성은 다른 결점과의 연결수가 많을수록 높아지는 값이다. 사이 중앙성은 비록 연결수가 많지 않을 경우라도 상이한 군집을 연결해주는 연계결점(mediating node)으로서의 역할을 한다면 네트워크에서 중요한 역할을 한다는 아이디어에 착안하여, 임의의 두 결점간에 최단거리에 위치하는 결점일수록 높은 값을 갖게 된다. 그리고 위세 중앙성이란 비록 스스로는 연결수가 많거나 연계결점으로서의 역할을 하지 못한다 할지라도 이러한 역할을 하는 결점과 직접적 연결관계에 있는 결점이라면 네트워크에서 중요한 영향력을 행사할 수 있기 때문에 이러한 결점들에 높은 값을 부여하는 측정 지표이다.

본 절에서는 네트워크에서 중요한 역할을 하는 결점들끼리 서로 연결되어 있는, 이른바 유유상종 현상이 나타나는지 분석하고자 한다. 각 결점의 연결정도 중앙성과 위세 중앙성간의 상관관계를 파악하여 상관계수가 높게 나오면 위세 중앙성이 높은 결점이 연결정도 중앙성 역시 높다는 것이므로 스스로 중요한 결점의 주변에도 중요한 결점이 위치한다는 것을 뜻하며 이는 결점들간의 유유상종 현상이 있다고 해석할 수 있는 것이다.

실제로 공개소프트웨어 커뮤니티의 590명의 개발자를 대상으로 연결정도 중앙성과 위세 중앙성간의 상관계수를 분석한 결과 0.81이라는 높은 값이 산출되었다. 이로써 공개소프트웨어 네트워크에 유유상종현상이 나타난다고 이해할 수 있다.

3.4

지금까지의 분석이 공개소프트웨어 네트워크의 위상구조적 특성을 분석한 것이라면 본 절에서는 네트워크의 위상구조적 특성과 결점이 지니고 있는 속성간의 관계를 파악하고자 한다. 우선 연결강도가 센 결점들간에 먼저 연결시키고 연결강도를 점차 낮춰가면서 나중에 연결시킴으로써 덴드로그램(dendrogram)을 작성한 후, 이를 적절한 수준에서 잘라내 네트워크를 위상구조적으로 근접한 다수의 결점으로 이루어진 여러 개의 하위 그룹, 즉 군집으로 구분한다. 그리고 각 군집을 구성하는 결점들의 속성을 조사하여 하나의 군집으로 묶인 결점들간에 나타나는 공통적인 속성, 즉 군집을 대표하는 속성을 파악한다. 이를 통해 공개소프트웨어 네트워크가 어떻게 하위구조로 분화되었으며, 분화된 그룹 내에서는 어떠한 결점의 속성을 공유하고 있는지, 그리고 상이한 군집 간에는 결점의 속성이 어떻게 차이가 나는지를 분석하고자 한다. 이 때 프로젝트 결점에 해당하는 개발자 결점이 있기 때문에 개발자 네트워크에서 하위그룹을 나누는 것과 프로젝트 네트워크에서 하위 그룹을 나누는 것은 동일한 분석이다.

따라서 본 절에서는 분석의 목적상 프로젝트 네트워크, 즉 프로젝트를 결점으로 하는 네트워크에서의 군집 분석을 하였다. 동일한 군집에 묶이는 프로젝트는 개발자를 공유하고 있기 때문에 이들 개발자들이 비슷한 특성과 관심을 갖고 있어 프로젝트의 속성 역시 비슷할 것으로 예상할 수 있으며, 군집을 구성하는 프로젝트 사이에 어떤 공통점이 있는지를 분석함으로써 개발자들의 프로젝트 참여 유인과 제약 등에 대해서 추론해 볼 수 있다.

KLDP 에서 운영되는 429개의 프로젝트를 군집으로 묶어보면 총 265개의 군집으로 묶을 수 있다. 이 중 하나의 결점이 하나의 군집으로 구성되는 것을 제외하고 2개 이상의 결점으로 구성되는 42개의 군집만을 분석의 대상으로 삼는다. 이 중 27개의 군집은 2개의 프로젝트로 구성되어 있는 소규모 군집이며 규모가 가장 큰 군집은 40개의 프로젝트로 구성되어 있다.

현재 KLDP 사이트는 환경(environment), 대상 사용자(intended audience), 소프트웨어 라이선스(license), 운영체제(operating system), 언어(language) 그리고 주제(topic) 등 6종류의 프로젝트 속성 정보를 제공해주고 있다. 군집 내에서 공통되는 프로젝트 속성을 요약해보면, 아래의 표와 같다. 예를 들어 대상 사용자가 누구인지를 기준으로 프로젝트의 속성을 구분해보면, 총 24개의 군집에서 군집 내의 프로젝트가 모두 동일한 대상 사용자를 위해서 개발되고 있다는 점에서 동질적이며, 이 중 14개의 군집은 일반사용자를 위한 소프트웨어만을 개발하는 프로젝트로 구성되어 있다.

본 절에서는 군집 내부의 프로젝트 간에 동질적 속성을 많이 보여주는 대상 사용자와 라이선스, 그리고 개발언어를 중심으로 네트워크의 위상구조적 특성과 프로젝트 속성간의 관계에 대해서 고찰하고자 한다.

Table. 1 Number of clusters that contains homogeneous project nodes

Project Attributes	# of Homogeneous Clusters(rates)	
Environment	15(0.36)	-console: 7, Windows: 4
Intended Audience	24(0.57)	-end users: 14, developers: 9
License	22(0.52)	-GPL: 19, BSD: 2
Operating	15(0.36)	-Linux: 5, Windows: 4

System		
Language	19(0.45)	-C/C++: 8, Python: 5, Java: 4
Topic	9(0.21)	-S/W develop.: 3, Internet: 2, game: 2

3.4.1

KLDP 커뮤니티에는 대상 사용자를 크게, 일반 사용자(end users/desktop), 개발자(developers), 시스템 관리자(system administrators), 그리고 기타(other audience)로 구분하고 있다. 누가 사용하는 프로그램을 개발하는지는 개발자가 어떤 분야의 소프트웨어 개발에 관심이 있는지와 개발자가 현재 어떤 위치에 있는지를 추론해볼 수 있다는 점에서 의미가 있다. 예를 들어 소프트웨어 개발업에 종사하는 참여자라면 평소 자신의 개발에 필요한 소프트웨어를 개발하고자 할 가능성이 높기 때문에 개발자를 대상으로 하는 공개소프트웨어 프로젝트에 참여할 것이다. 실제로 KLDP의 전체 프로젝트 중 43%는 개발자를 대상으로 하는 프로그램을 개발하고, 40%는 일반 사용자를 대상으로 하는 프로그램을 개발하고 있어, 큰 차이를 나타내지 않고 있다. 그렇지만 일반사용자를 대상으로 하는 프로그램이 예상보다 많이 개발되고 있다는 점을 알 수 있다.

또한 전체 42개의 군집 중에서 대상 사용자를 프로젝트 속성으로 삼아 동일 군집에 속하는 프로젝트 간의 동질성 수준을 조사해보면 전체 42개의 군집 중 57%에 해당하는 24개의 군집이 동질적 속성을 가지는 프로젝트로만 구성이 되었음을 파악했다. 이 중 14개의 군집은 일반사용자를 위한 소프트웨어 개발 프로젝트만으로 구성되어 있고, 9개의 군집은 개발자를 위한 프로그램만을 개발하는 프로젝트만으로 구성되어 있다. 일반사용자를 대상으로 하는 프로젝트의 수와 개발자를 대상으로 하는 프로젝트의 수가 비슷함에도 불구하고 일반사용자를 대상으로 하는 프로젝트만으로 구성된 군집이 상대적으로 더 많다는 것은 일반사용자를 위한 프로젝트를 개발하는 사람들은 개발자용 프로그램의 개발에 참여하는 빈도가 낮음을 의미한다. 즉, 일반사용자를 대상으로 하는 프로그램에 특화되어 참여한다.

3.4.2

Lerner and Tirole(2005)에 의하면 오픈소스 라이선스는 상용화 허용 정도에 따라 허용적인 라이선스(permissive license), 제한적인 라이선스(restrictive license), 그리고 매우 제한적인 라이선스(highly restrictive license)로 구분할 수 있다. 허용적인 라이선스 중에서 가장 많이 사용되는 라이선스는 BSD(Berkeley Software Distribution) 라이선스이고, 제한적인 라이선스는 LGPL(“Lesser” General Public License), 그리고 매우 제한적인 라이선스는 GPL(GNU General Public License) 라이선스이다.

GPL라이선스는 프로그램을 소스코드의 형태로 양도하면서 이에 대한 이용, 복제, 수정 및 배포의 자유뿐만 아니라 수정된 프로그램에 대해서도 동일한 자유가 계속해서 보장되어야 하기 때문에 라이선스 요금 기반의 소프트웨어 사업이 불가능하다. 반면 BSD라이선스는 공개소프트웨어의 상업적 이용에 제약이 없으며 사적 재산으로 보호되는 2차적 저작물의 개발도 가능하며, 대부분의 상용화 가능 라이선스는 BSD라이선스를 일부 수정한 것들이 대부분이다.

Lerner and Tirole (2005)는 SourceForge 사이트를 조사하여 개발되는 소프트웨어의 특성 및 대상 사용자의 특성과 선택한 공개소프트웨어 라이선스와의 인과관계를 분석하였다. 또한 Fershtman and Gandai(2004)은 허용적인 라이선스를 채택한 프로젝트에서 개발자 일인당 기여수준이 제한적 라이선스에 비해서 더 높지만 프로젝트에 참여하는 개발자의 숫자는 제한적 라이선스를 채택한 프로젝트가 더 많음을 분석했다. 이 외에도 공개소프트웨어 라이선스로서 어떤 라이선스가 바람직한지에 대한 다수의 논문이 존재한다.(Tsur and David, 2004; Evans and Reddy, 2002; Hahn, 2002) GPL과 같은 제한적 라이선스를 지지하는 연구는 진정한 공개소프트웨어가 되기 위해서는 상용화를 목적으로 둔 개발자들에 의해서 무임승차되지 않아야 하며, 지속적인 협업을 이끌어낼 수 있는 메커니즘이 라이선스에 포함되어야 한다는 논리를 주장하고, BSD, Apache, MIT/X Consortium 라이선스와 같은 상용화 가능 라이선스를 지지하는 연구는 상용화를 가능하게 함으로써 경제적 유인을 가진 개발자들 역시 공개 소프트웨어 커뮤니티에 자신의 리소스를 투여할 수 있도록 해야 공개소프트웨어가 궁극적으로 성장할 수 있다고 논리에 근거하고 있다.

실제로는 공개소프트웨어에 대한 사회적 관심을 불러일으킨 Free Software Foundation에서 제시한 GPL 라이선스가 공개소프트웨어의 상징적 라이선스로 자리 잡고 있으며, 가장 성공적인 공개소프트웨어인 리눅스 역시 GPL 라이선스를 채택하고 있어 대부분의 공개소프트웨어는 GPL을 따르고 있다. KLDP에서 운영되는 프로젝트 중 93%는 OSI(Open Source Initiative)에서 인증 받은 라이선스를 채택하고 있다. 이 중 GPL 라이선스를 채택한 프로젝트는 314개이고, LGPL과 BSD는 각각 42개씩의 프로젝트가 운영 중에 있다. 기타 MIT/X 라이선스(10개), Zlib/libpng 라이선스(2개), Python라이선스 등은 모두 허용적인 라이선스에 속하며, OSI의 인증을 받지 못했지만 Public Domain 라이선스 역시 상용화에 제한이 없다는 점에서는 허용적인 라이선스(10개)라고 할 수 있다.

앞서 밝힌 바와 같이 라이선스 결정 문제는 개발자들이 소프트웨어를 어떤 목적으로 개발하는지가 반영이 되어 있을 뿐만 아니라 공개소프트웨어란 무엇인가에 대한 개발자의 규범적 가치관이 반영되어 있다. 따라서 본 연구에서는 80%에 가까운 대부분의 프로젝트가 GPL이나 LGPL과 같이 상용화에 제한을 둔 라이선스를 채택하고 있기 때문에 소수의 상용화 가능한 라이선스를 채택하고 있는 프로젝트가 균집을 이루고 있는지 아니면 다수의 균집에 분산되어 있는지를 파악하고자 했다.

예를 들어 상용화 가능 라이선스로만 구성된 균집이 다수 존재한다면 실제로 KLDP 커뮤니티는 상용화가 가능한 라이선스 균집과 상용화가 불가능한 라이선스 균집으로 구분된다고 볼 수 있으며 각각의 라이선스를 지지하는 개발자 사이에는 협업이 어려울 것이라고 해석할 수 있다.

분석 결과 전체 균집의 52%에 해당하는 22개의 균집이 동질적인 라이선스를 채택하고 있는 프로젝트로 구성되어 있음을 확인했다. 하지만 이 중 19개의 균집은 GPL 라이선스를 채택한 프로젝트만으로 구성되어 있으며, BSD 라이선스를 채택한 프로젝트만으로 구성된 균집은 2개에 불과했다. 하지만 2개 이상의 이질적인 라이선스를 선택하고 있는 균집을 포함하는 전체 42개의 균집에 속하는 프로젝트를 대상으로 조사해보면 균집 내에서 BSD 라이선스를 채택하고 있는 20개의 프로젝트 중 14개의 프로젝트를 구성하고 있는 8명의 개발자들은 GPL이나 LGPL과 같이 제한적인 성격의 라이선스를 채택한 프로젝트에 참여하는 빈도가 매우 낮음을 발견했다. 전체 프로젝트에서 GPL/LGPL 라이선스를 채택한 프로젝트

비율이 80%라는 점을 고려하고 아래의 Table.2를 보면 이들이 상용화 가능 라이선스를 선호할 뿐만 아니라 라이선스 종류가 이들이 프로젝트 참여 여부를 결정하는데 중요한 영향을 미친다고 추론해 볼 수 있다.

Table. 2 Developers' participation rates of projects that choose restrictive licenses

Developer ID	L/GPL Participation Rates	Developer ID	L/GPL Participation Rates
D138	0/4 (0)	D395	0/3 (0)
D142	1/3 (0.33)	D416	2/5 (0.4)
D371	1/3 (0.33)	D471	1/4 (0.25)
D390	0/6 (0)	D550	3/8 (0.38)

3.4.3

개발언어는 개발하고자 하는 프로그램의 성격과 운영환경, 그리고 개발자의 개발 능력을 반영하여 결정된다. 이 중에서 개발자의 개발 능력이라 함은 개발자들이 선호하거나 전문화되어 있는 개발 언어를 의미한다. 실제로 KLDP 커뮤니티에서 개발자들이 작성하게 되는 개인별 스킬 프로필에도 전문성을 갖춘 개발 언어를 표시하도록 되어 있다.

군집 조사 결과 전체 군집 중 45%에 해당하는 19개의 군집이 동일한 개발언어를 사용하는 프로젝트로 구성되어 있었다. 하지만 동질적인 언어를 사용하는 군집의 경우 이들 개발언어와 다른 프로젝트 속성들-환경, 운영체제, 주제-간의 상관관계를 발견할 수 없었다. 따라서 개발언어는 개인적인 선호나 개발 능력을 반영한다고 생각할 수 있다. 19개의 동질적인 군집 중 8개는 C/C++언어로 개발되고 있으며, Python은 5개의 군집, Java는 4개의 군집에서 동질적으로 사용되고 있는 것으로 보아 개발언어로 가장 널리 활용될 뿐만 아니라 개발자들 사이에서 가장 강한 충성도가 높은 개발언어로 나타났다.

4.

본 연구는 공개소프트웨어 커뮤니티에 가입한 개발자들이 커뮤니티 안에서 어떠한 인적 네트워크를 구성하여 소프트웨어 개발에 참여하는지를 파악하는 것을 목적으로 네트워크의 위상구조적 특성을 분석했다. 분석 결과 다음과 같은 네트워크 특성을 지니고 있음을 확인했다. 첫째, 공개소프트웨어 네트워크는 일인당 평균 1.35개의 프로젝트에 참여하고 전체의 65%에 해당하는 277개의 프로젝트가 한 명의 개발자에 의해서 운영되는 등 개발자들간에 연결 수가 많지 않기 때문에 네트워크 밀도가 0.006수준으로 다른 사회 네트워크에 비해서 상당히 낮은 수준이다. 둘째, 공개 소프트웨어 프로젝트 규모 분포와 일인당 참여하는 프로젝트의 수 분포에 있어서 멱함수 법칙이 성립했다. 이는 개발자들이 규모가 큰 프로젝트에 참여함으로써 비교적 적은 수의 프로젝트에 참여하고도 다수의 개발자들과 연결관계를 맺게 됨을 의미한다. 셋째, 공개소프트웨어 커뮤니티는 밀도는 낮지만 결속계수는 매우 높음을 확인했다. 이는 네트워크의 전반적인 연결수준은 낮지만

일단 특정 프로젝트에 참여하면 동일 프로젝트 참여자와 모두 연결관계를 갖게 되어 국소적인 연결수준이 높아지는 공동참여네트워크의 특성을 반영하는 것으로 해석할 수 있다. 넷째, 네트워크에서 중요한 위치를 차지하는 개발자는 역시 네트워크에서 중요한 위치에 있는 개발자와 연결되어 있어 유유상종 현상이 나타났다. 즉 연결수 중앙성이 높은 개발자들이 위세 중앙성 역시 높게 나타나는 것을 통해서 확인할 수 있었다. 다섯째, 프로젝트 네트워크를 군집으로 묶었을 때 동일한 군집에 묶이는 프로젝트는 동일한 개발자를 공유하고 있기 때문에 비슷한 속성을 지니는 경우가 많은데 특히 대상사용자와 라이선스, 그리고 개발 언어가 동질적인 클러스터가 많이 관찰되었다.

본 연구는 다음과 같은 점에서 연구의 한계와 향후 확장 가능성이 있다. 우선 공개소프트웨어 커뮤니티의 네트워크가 지니고 있는 구조적 특성을 파악하는데 초점을 맞추었기 때문에 이러한 특성이 나타나게 된 원인을 파악하기 위해서는 커뮤니티 참여자를 대상으로 한 설문 조사 등 추가적인 분석 작업이 필요하다. 둘째, 본 연구는 국내에서 운영되는 단일 공개소프트웨어 커뮤니티를 대상으로 분석했기 때문에 본 연구에서 분석한 결론을 모든 공개소프트웨어 커뮤니티에 적용하여 일반화 하는 것에는 한계가 있으며 다른 공개소프트웨어 네트워크를 분석한 연구와 비교함으로써 공개소프트웨어 커뮤니티의 공통적 특성과 커뮤니티의 고유한 특성을 구분해 낼 수 있을 것이다. 셋째, 본 연구는 공개소프트웨어 개발 커뮤니티의 네트워크 속성을 파악하는데 초점을 맞추었기 때문에 커뮤니티의 동태적 분석을 통해 향후 예측 및 커뮤니티 활성화를 위한 방안을 제안해주는 확장 연구의 필요성이 있으며, 이러한 작업은 행위자 기반 시뮬레이션 기법 등을 통해서 보완이 될 수 있다고 생각한다.

1. A. L. Barabasi and R. Albert, "Emergence of Scaling in Random Network", Vol. 286, Science, pp.509-512(1999).
2. D. Evans, B. Reddy, "Governmental Preferences for Promoting Open Source Software: A Solution in Search of A Problem", National Economic Research Associates (2002).
3. C. Fershtman, N. Gandal, "The Determinants of Output per Contributor in Open Source Projects: An Empirical Examination" CEPR Working Paper 2650 (2004)
4. R. Hahn, "Government Policy toward Open Source Software", AEI-Brooking Joint Center for Regulatory Studies (2002).
5. A. Khalak, "Agent-based Model for Economic Impact of Free Software", Complexity 8, pp. 45-55, (2003).
6. J. Lerner, J. Tirole, "Some Simple Economics of Open Source", Vol. 50, No. 2, Journal of Industrial Economics, pp. 197-234, (2002).
7. J. Lerner , J. Tirole, "The Scope of Open Source Licensing", Vol. 21, No. 1, Journal of Law, Economics and Organization pp. 20-56 (2005).
8. G. Madey, V. Freh, R. Tynan, "Agent-based Modeling of Open Source Using Swarm", 8th Americas Conference on Information Systems, (2002).
9. G. Navarro, "Self-Organization in Open Source Communication Networks", International Journal of Information Ethics Vol. 2, pp.1-9 (2004).
10. M. Tsur, S. David, "A License to Kill Innovation: On Open Source Licenses and Their Implications for Innovation", Working Paper (2005).
11. S. Valverde, G.. Theraulaz, J. Gautrais, V. Fourcassie, R. Sole, "Self-Organization Patterns in Wasp and Open Source Communities", IEEE Intelligent Systems, 36-40 (2006).
12. S. Weber, "The Success of Open Source", Harvard University Press (2004).
13. J. Xu, Y. Gao, S. Christley, G.. Madey, "A Topological Analysis of the Open Source Software Development Community", Proceedings of the 38th Hawaii International Conference on System Sciences (2005).
14. , , " ", (2006).

Analyzing Open Source Software Community: Topological Properties and Its Implications

Won-Young Cho[†], Dae-Chul Jang^{*}, and Byong-Hun Ahn^{**}

[†]KAIST Graduate School of Management

^{*}KAIST Graduate School of Management

^{**}KAIST Graduate School of Management

Abstract

Open source software(OSS), which is distributed with source code for free and anyone can redistribute and modify without permission of licensor is one of the most prominent phenomenon in the software industry. Therefore, more and more developers participate in OSS Internet communities and produce OSS . These communities are generally complex adaptive network. This paper analyzes the topological properties of KLDP site which is one of the largest OSS development communities in Korea. We find that the number of links is small but the network has some characteristics that can often be observed in complex network such as power law. We also analyze the relationship between the topological properties of software project and its own attributes and then find that a lot of attributes are homogeneous among the projects which are linked directly. Other topological analysis such as clustering coefficient and homophily phenomenon are also presented.

Key Words : open source software, network topology, complex system, internet community