
소프트웨어 변경가시성의 의존도분석을 이용한 형상관리

김경훈[†], 경태원^{††}, 송영재^{†††}

Configuration Management Using Dependency Analysis of Software Change Visibility

Kyoung-Hun Kim[†], Tae-Won Kyung^{††}, Young-Jae Song^{†††}

요약 소프트웨어 형상관리는 소프트웨어 제품에 기반이 되는 대규모 소프트웨어 산출물 집합을 관리하는 분야이다. 형상관리는 과거 수년 동안 빠르게 발달해왔지만, 새로운 형상관리 시스템은 여전히 요구되고 있다. 전통적인 방법을 쓰거나 형상관리에 대한 연구보다는 현재 개발 프로젝트의 필요한 전문적인 관리가 필요하다. 따라서 형상관리에 필요한 저장소의 일반적인 모델이나 형상관리에 변형에 따른 과정을 관리하고 프로그램에 연결되어 있는 프로그램의 인터페이스를 분리하여 관리하는 기법이 필요하다. 본 논문은 형상관리에 필요한 저장소를 제공하고 프로그램과 인터페이스를 분리하여 관리하는데 필요한 의사결정 시스템을 도입하였다. 따라서 필요한 형상관리의 의존도를 가지고 새로운 접근법을 시도하였다. 의존도가 높은 개별적 컴포넌트의 변형과 결합을 위해 의사결정 시스템을 이용하여 확인 및 검증을 하여 소프트웨어 개발에 필요한 형상관리를 제공하고 있다.

주제어 : 컴포넌트, 의존도, 네트워크분석 기법

Abstract Software configuration management is a field of managing sets of large scale software outputs to be the basis of the software products. Configuration management has significantly improved for the past few years, yet new system is still required. Instead of using previous methods or researching about configuration management, professional management of developing project is needed. Therefore, techniques such as storage model for configuration management, management program for change procedure and management of program interface is needed. In this paper, we provide storage for configuration management and implement decision-making system to manage divided program and interface. Thus, we propose a new access method by the configuration management dependency. This paper provides configuration management for developing software to verify and classify using decision-making system for the change and combination of independent components with high dependency.

Key Words : Configuration management, Component, ANP

1. 서론

1.1 형상관리

소프트웨어 형상관리는 소프트웨어 시스템의 개발과 유지보수에 있어 핵심적인 분야이다[1][2][3][4]. 소프트웨어 개발의 특징은 사용자로부터의 요구사항이 계속해서 변한다는 점이며, 개발자 혹은 개발팀에서 또한 기술

적인 면에서도 수정이 불가피하게 된다. 즉, 소프트웨어 개발 프로젝트 과정에서 여러 문서들이 만들어 지며, 또한 여러 가지의 버전이 존재하게 된다. 따라서 형상관리는 시스템의 변경을 제어하고 진행 중인 프로젝트의 여러 버전을 관리하는 프로세스이다. 따라서 형상관리는 소프트웨어 개발 및 유지보수를 위해 구조를 통제하고 변경요구를 관리하는 프로세스로서의 역할을 필요로 한다.

[†] 경희대학교 컴퓨터공학과 박사수료 / 강동대학교 교수

^{††} R&D 특허센터 선임연구원 (교신저자)

논문접수 : 2012년 1월 18일, 1차 수정을 거쳐, 심사완료 : 2012년 2월 17일

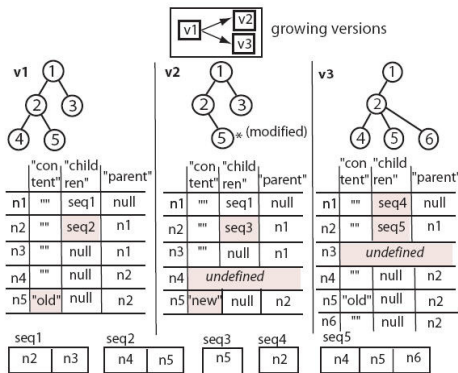
1.2 기존의 형상관리

일반적인 형상관리는 파일-기초의 버전 컨트롤 방법으로 소스코드에 중점을 두었기 때문에 확장 및 변경에 대한 모듈을 선택하기 어려운 단점이 있다. 즉, 추상계층에 대한 정보가 없고 독립적이지 못하기 때문에 새로운 버전에 대한 관리를 지원하지 못하고 있다[5][6].

Fine-grained 데이터 모델을 사용한 변경관리는 UML 다이어그램에 요소들로 그림1에 보이는 것처럼 계층을 표시하여 관리한다. 소프트웨어 형상항목을 이용하여 계층간의 상속, 결합관계 등을 표현하지만 구조를 분석하는데 많은 노력이 필요하며 새로운 버전에 대한 변화에 적응 못하는 단점이 있다[7][8].

변경관리에서 초기 작업은 어플리케이션의 구조의 저장과 구조의 효과를 탐색하기위한 변화와 허가를 위해 제공된 프레임워크에 의해서 결과물의 주소를 할당한다 [9][10][11]. 그러나 네트워크 관리에 대해서는 지원을 하지 않는 단점이 있다.

현재 개발되는 프로젝트 대부분은 분산환경을 제공하고 있고 또한 실시간 변화를 갖기 때문에 분산환경과 실시간 변화를 지원하는 형상관리가 필요하다.



[그림 1] Fine-grained 관리 모델

1.3 컴포넌트 형상관리

컴포넌트 형상관리는 컴포넌트간의 복잡한 상호작용은 수용할 수 있는 변경을 정의하고, 정의된 범위로 부터의 이탈을 위한 변경을 감시하고, 그리고 이탈의 수행을 위한 변경을 관리하는 것이다.

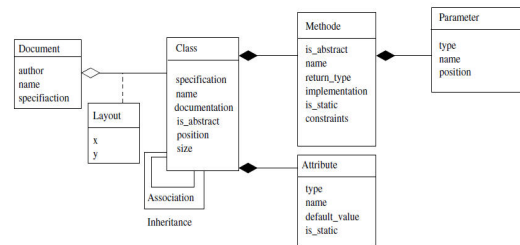
소프트웨어 개발 과정에서 생성된 산출물들을 인위적으로 관리하고, 소프트웨어의 변화들과 컴포넌트들의 관

계를 조정하고, 버전을 저장하여 소프트웨어 개발에 도움을 준다.

[그림 2]는 컴포넌트 데이터 모델의 관계를 객체의 의존도와 명세에 대한 관계를 나타내고 있다. 그러나 새로운 컴포넌트의 추가될 경우 각 컴포넌트에서 객체와 멤버 함수 및 변수에 대한 의존도와 명세를 추가하여야한다. 모든 관계에 대한 고려에 의해 복잡성이 증가하여 새로운 컴포넌트에 대한 요구사항이 증가하게 된다.

소프트웨어 개발 기반의 컴포넌트(Component-based software development[CBS])의 목적은 새로운 소프트웨어 시스템을 미리 만들어진 소프트웨어를 재사용하여 개발하기 위한 것이다. 컴포넌트를 재사용하기 위해 구조기술언어(ADS)가 시스템 모델링의 능력을 높이기 위해 형상관리 시스템으로서의 통합을 제한했다. 그러나 소프트웨어 구조상 새로운 시스템을 구성할 때 직접적으로 다른 컴포넌트들과 연결하기 위한 컴포넌트의 재사용은 감소하고 있다. 의존도의 평가 없이 재사용되어진 컴포넌트는 결합도가 약하여 최적의 성능에 부적합하다.

따라서 본 논문은 의존도의 평가와 결합도를 위해 ANP(Analytic Network Process)기법을 이용하여 서로 다른 컴포넌트에 재사용에 적합하도록 하였다.



[그림 2] 컴포넌트 데이터 모델

2. 퍼지 네트워크 분석기법

본 연구에서는 컴포넌트간의 의존관계 측정을 위해 설계 지표들의 가중치와 우선순위를 측정하기 위해 퍼지 네트워크 분석기법(Fuzzy Analytic Network Process)을 이용하였다.

ANP는 가장 널리 쓰이는 다기준 의사결정기법 중 하나인 AHP의 보다 일반화된 형태로 Saaty에 의해 개발되었다[9][10]. AHP는 하나의 문제를 여러 계층구조를

가지는 다수의 수준으로 분해하고 각 수준에 존재하는 각 의사결정 요소들의 상호 독립적이라고 가정하고 있다. 그러나 ANP는 이러한 AHP를 의사결정 요소간 의존과 피드백을 내포하는 다른 문제로 확장시킨다. 이와 같은 개념의 확장은 문제 내에 존재하던 계층 구조를 네트워크 구조로 대체함으로써 인해 의사결정 요소들간의 복잡한 상호 관계성을 포함하게 만든다[11].

본 연구에서는 ANP기법의 퍼지 이론을 적용하여 컴포넌트 선택 관계를 쌍대비교의 결과를 통해 상대 중요도를 도출하여 고유벡터법을 이용하여 의존도를 도출하였다.

쌍대비교를 위한 퍼지 적용 과정은 다음과 같다. 퍼지 쌍대비교 행렬 A가 다음과 같다고 가정하면,

$$A = [a_{ij}] = [(l_{ij}, m_{ij}, u_{ij})],$$

$$(i, j = 1, 2, \dots, n)$$

$$= \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

$i = j$ 인 모든 $a_{ij} = (1, 1, 1)$ 이고,

$l_{ij} = \frac{1}{l_{ji}}, m_{ij} = \frac{1}{m_{ji}}, u_{ij} = \frac{1}{u_{ji}}$ 의 관계가 성립

하고, 퍼지 AHP 적용 절차는 다음과 같다.

1단계, i 번째 요소의 Fuzzy Synthetic Extent 값을 E_i 라 하면, E_i 는 다음과 같이 정의된다.

$$E_i = \sum_{j=1}^n a_{ij} \otimes \left(\sum_{i=1}^n \sum_{j=1}^n a_{ij} \right)^{-1} \quad (\text{식 1})$$

2단계, 삼각 퍼지수

$M_1 = (l_1, m_1, u_1), M_2 = (l_2, m_2, u_2)$ 에 대하여 $M_2 \geq M_1$ 일 확률의 정도는 다음과 같이 정의된다.

$$V(M_2 \geq M_1) = \text{hgt}(M_2 \cap M_1) = \mu_{M_2} \quad (d)$$

$$\begin{cases} 1, & \text{if } m_2 \geq m_1 \\ 0, & \text{if } l_1 \geq u_2 \\ \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, & \text{otherwise} \end{cases}$$

(식 2)

여기서 d 는 u_{M_1} 과 u_{M_2} 의 교차점의 x 좌표 값을 의미한다.

3단계, 삼각 퍼지수 M_i 가 다른 k 개의 퍼지수

$M_i (i = 1, 2, \dots, n)$ 보다 클 확률의 정도는 다음과 같이 정의된다.

$$V(M \geq M_1, M_2, \dots, M_k)$$

$$= V[(M \geq M_1) \text{ and } (M \geq M_2) \text{ and } \dots \text{ and } (M \geq M_k)]$$

$$= \min V(M \geq M_i), i = 1, 2, \dots, k.$$

(식 3)

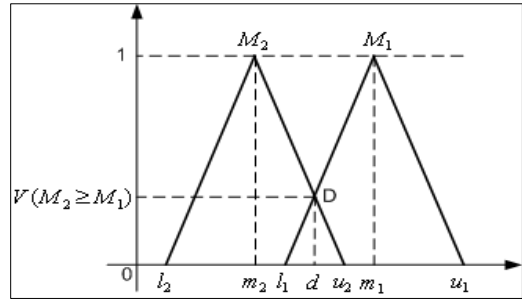
4단계, 특정 요소 $i (i = 1, 2, \dots, n)$ 에 대하여

$$w' = \min V(E_i \geq E_j),$$

$$(j = 1, 2, \dots, n; i \neq j)$$

라고 가정하면, 각 요소들의 가중치 벡터는 다음과 같다.

$$W' = (w'_1, w'_2, \dots, w'_n)^T \quad (\text{식 4})$$



[그림 3] M1과 M2의 교집합

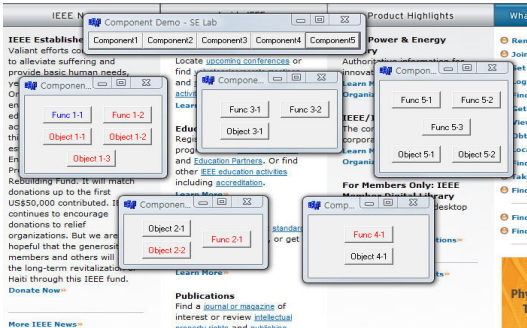
그리고, 이를 정규화하면 다음과 같은 각 요소들의 정규화된 가중치 벡터 W 를 구할 수 있다.

$$W = (w_1, w_2, \dots, w_n)^T \quad (\text{식 5})$$

본 연구에서는 Chang의 알고리즘을 적용하여 쌍대비교 과정에서 선택된 컴포넌트들에 의존도를 명확하게 계량화시킴으로서 상대적 중요도를 비교하였다.

3. 네트워크 분석기법을 위한 네트워크 모델 설계

본 연구에서 컴포넌트간의 재사용과 실제 멤버함수와 변수의 사용에 대한 측정을 위해 모델을 생성하고 서로의 의존도를 측정하였다.



[그림 4] 컴포넌트의 의존관계

[그림 4]는 윈도우 환경에서 컴포넌트들에 의존관계를 보여주고 있다. 각각에 요소들이 참조하는 모습과 선택된 요소의 값을 선택적으로 보여준다.

실제 컴포넌트에서 다른 컴포넌트의 요소에 참조하는 내용을 분석하였다.

```

__fastcall TComponent1::TComponent1(TComponent* Owner)
: TForm(Owner)
{
}

void __fastcall TComponent1::Init(TSpeedButton *OBJ) {
    Main_Component->Clear();
    OBJ->Font->Color = c1Blue;
}

void __fastcall TComponent1::Call(TSpeedButton *OBJ) {
    OBJ->Font->Color = c1Red;
}

void __fastcall TComponent1::Func1Click(TObject *Sender)
{
    Init((TSpeedButton *)Sender);

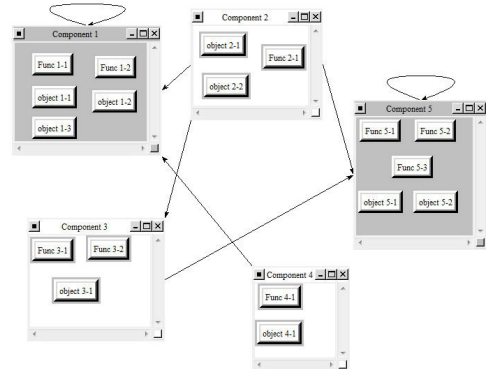
    Call(Func2);
    Call(Object1);
    Call(Object2);
    Call(Object3);
    Call(Component2->Func1);
    Call(Component2->Object2);
    Call(Component4->Func1);
}
    
```

[그림 5] 참조관계 분석 코드

[그림 5]는 컴포넌트간에 참조되는 형태를 표현해주는 코드이다. 코드를 통해 서로 상호 참조관계를 그래픽적으로 나타낸 것이 [그림 4]이다.

4. 의존관계 모형 및 분석

의존도측정 알고리즘을 통해 의존관계 테이블을 작성하고 테이블 자료를 바탕으로 의존 관계 네트워크 모형을 [그림 6]으로 설계하였다.



[그림 6] 의존 관계 모형

5. 결론 및 향후 연구과제

컴포넌트들의 의존도 요인, 의존도 분석, 그리고 매트릭스 기법을 사용하여 단계별 점수로 환산하여 매트릭스로 의존도를 계산하였다.

	변값	연값	연값	연값	연값	연값	연값
변값	1.0000	0.2	8	2	2		
연값	5.0000	1.0000	5	6	7		
연값	0.1250	0.2000	1.0000	0.2	0.2		
연값	0.5000	0.1857	5.0000	1.0000	0.5		
연값	0.5000	0.1429	5.0000	2.0000	1.0000		
연값							
연값							

[그림 7] 의존도의 가중치 계산

[그림 7]은 의존관계 모형으로 가중치를 계산하여 컴포넌트간의 의존도를 계산한 테이블값이다.

Normalized Matrix					Weights / Rank		Product / Ratios		CI / RI
0.1404	0.1224	0.2963	0.1786	0.1869	0.1849	2	1.0083	5.4533	CI : 0.0926 RI : 0.1597
0.7018	0.6118	0.2963	0.5357	0.6542	0.5600	1	3.2053	5.7243	
0.0175	0.0765	0.0370	0.0179	0.0187	0.0335	5	0.1710	5.0999	
0.0702	0.1020	0.1852	0.0893	0.0467	0.0987	4	0.5135	5.2046	
0.0702	0.0874	0.1852	0.1786	0.0935	0.1230	3	0.6603	5.3705	

[그림 8] 의존도 계산 결과

본 연구에서는 실제 컴포넌트간의 의존도를 계산하여 관리할 수 있는 플랫폼을 제공하였다. 향후 연구에서는 클라우드를 이용하여 실시간 동기화 가능한 플랫폼에 대한 연구를 진행 중이다. 실제 소프트웨어 개발환경을 클라우드 시스템에 적용하여 새로운 변화에 적응 할 것으로 기대된다.

참 고 문 헌

- [1] S. Dart., (1991). Concepts in Configuration Management Systems, Proc. Third Int'l Software Configuration Management Workshop, pp. 1-18, June.
- [2] U. k. Ministry of Defence.(2000). Configuration Management of Defence Material. Defence Standard 05-57/Issue 4, July.
- [3] E. H. Bersoff, V. D. Henderson, and S.G. Siegel.(1980) *Software Configuration Management*, Prentice Hall.
- [4] D. Whitgift.(1991) *Methods and Tools for Software Configuration Management*. John Wiley and Sons.
- [5] B. W. Jhonson.(1989). *Design and Analysis of Fault-Tolereant Dgital System* Addison -Wesley.
- [6] D.K. Pradhan.(1996). *Fault-Tolerant Computer System Design*, Prentice-Hall.
- [7] Dirk Ohst, Udo Kelter.(2002).A Fine-grained Version and Configuration Model in Analysis and Design. ICSM, IEEE.
- [8] Tien N. Nguyen, Ethan V.Munson, John T.Boyland.(2005). Multi-level Configuration Management with Fine-grained Logical Units. EUROMICRO-SEAA'05.
- [9] Saaty, T. L.(1980).The Analytic Hierarchy Process. McGraw-Hill, New York.
- [10] Saaty, T. L. (1996). Decision making with dependence and feedback: The analytic network process. Pittsburgh:RWS Publications.
- [11] Meade, L., Sarkis, J. (1999). Analyzing organizational project alternatives for agile manufacturing processes: An analytic network

approach. International Journal of Production Research, 37(2), pp. 241-261.

김 경 훈



- 2000년 : 삼육대학교 컴퓨터과학과 (이학사)
- 2002년 : 경희대학교 전자계산 공학과(공학석사)
- 2004년 : 경희대학교 전자계산공학과 박사 수료
- 2012년 ~ 현재 : 강동대학교 교수
- 관심분야: 형상관리, 웹서비스, 의료시스템, 콘텐츠
- E-Mail: magiclamp@khu.ac.kr

경 태 원



- 1998년 : 호원대학교 전자계산학과 (이학사)
- 2002년 : 경희대학교 전자계산공학과(공학석사)
- 2008년 : 경희대학교 산업공학과(공학박사)
- 2008년 ~ 2010년 : 한국생산기술연구원

구원

- 2010년 ~ 현재 : R&D특허센터
- 관심분야 : IT 프로젝트 기획/관리, 전자연구노트 시스템
- E-Mail : twkyung@rmdip.re.kr

송 영 재



- 1969년 : 인하대학교 전자공학과(공학사)
- 1976년 : 일본 keio 대학교 전산학과 (공학석사)
- 1980년 : 명지대학교 전산학과(공학박사)
- 1982년 ~ 1983년 : 미국 Maryland

University 전산학과 연구교수

- 1989년 ~ 1990년 : 일본 Keio University 전산학과 객원교수
- 1976년 ~ 현재 : 경희대학교 컴퓨터공학과 교수
- 관심분야 : 소프트웨어 재사용, CASE 도구, AOP, 요구공학, 컴포넌트웨어, 웹서비스
- E-Mail : yjsong@khu.ac.kr