

# 자가 적응 시스템에서의 목표 모델의 동적 가중치 변경에 관한 연구

황다솜, 이종현, 이은석  
성균관대학교 정보통신공학부

e-mail : [rhcqnspp32@skku.edu](mailto:rhcqnspp32@skku.edu), [carve9142@skku.edu](mailto:carve9142@skku.edu), [leees@skku.edu](mailto:leees@skku.edu)

## A Study on dynamic weight-changing method of goal model for self-adaptive system

Dasom Hwang, Chonghyun Lee, Eunseok Lee  
Dept. of Information and Communication Engineering, Sungkyunkwan University

### 요 약

자가 적응 시스템은 사람의 직접적인 개입 없이 자율 제어를 통한 자가 최적화 (self-optimization), 자가 치유 (self-healing) 등의 능력이 요구되고, 이러한 시스템은 시스템이 조달된 환경과 시스템 내부 상황을 고려한 적절한 적응 정책과 목표 평가를 통해 시스템의 신뢰성을 보장할 수 있어야 한다. 목표 기반의 자가 제어 시스템은 목표 만족도에 따라 시스템을 자율 제어하기 때문에 목표 기반 자가 적응 시스템에서의 목표 만족도(goal satisfaction) 평가는 매우 중요하지만 기존의 연구들의 목표 만족도 평가 방법에서는 환경 변화가 반영되지 않는다는 한계가 있다.

본 논문에서는 목표 모델에서의 상위 목표에 대한 하위 목표들의 기여도에 따라 가중치를 부여하고 시스템의 외부 환경 변화에 따라 가중치를 동적으로 변경하는 방법을 제안한다. 이를 통해 기존의 목표 평가 방법보다 사용자의 요구가 잘 반영되고 신뢰성 높은 평가가 가능하다.

### 1. 서론

자가 적응 시스템은 사람의 직접적인 개입 없이 자율 제어를 통한 자가 최적화(self-optimization), 자가 치유 (self-healing) 능력 등이 요구된다[1]. 최근 미션-크리티컬 (mission-critical) 시스템에서도 자율 제어가 가능하도록 많은 연구가 진행되고 있다. 이러한 시스템은 시스템이 조달된 환경과 시스템 내부 상황을 고려한 적절한 적응 정책으로 시스템의 신뢰성을 보장 할 수 있어야 한다.

목표 모델은 사용자의 관점에 가깝게 요구 사항을 도출하기 때문에 시스템에 필요한 기능들을 설계하는데 용이하고 목표를 기반으로 시스템 스스로 자신의 임무 수행 능력을 평가하는 것이 가능하므로 최근에 목표 기반 (goal-oriented) 자가 적응 시스템 설계 기법들이 제안되고 있다[2][3][4]. 목표 기반의 자가 적응 시스템에서는 목표 만족도에 따라 시스템을 자율 제어하기 때문에 목표 만족도(goal satisfaction) 평가는 매우 중요하다. 하지만 기존의 연구들의 목표 만족도 평가 방법에서는 환경 변화가 반영되지 않는다는 한계가 있다.

본 논문에서는 목표 모델에 하위 목표들 간의 상대적 중요성과 상위 목표에 대한 기여도에 따라 가중치를 부여하고 시스템의 외부 환경 변화에 대해 가중치를 동적으로 변경하는 방법을 제안한다. 본 연구 결과를 통해 환경 변화를 반영한 목표 평가가 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련이 있는 배경 연구를 살펴보고, 3장에서는 본 논문에서 제안하는 동적 가중치 변경 방법을 설명한다. 본 연구에서는 공간 감시 로봇을 이용한 시뮬레이션을 통해 제안 방법의 유효성을 검증하였으며, 실험에 관련된 내용은 4장에 작성하였다. 마지막으로 5장에서는 결론과 향후연구에 대해 기술하며 마무리한다.

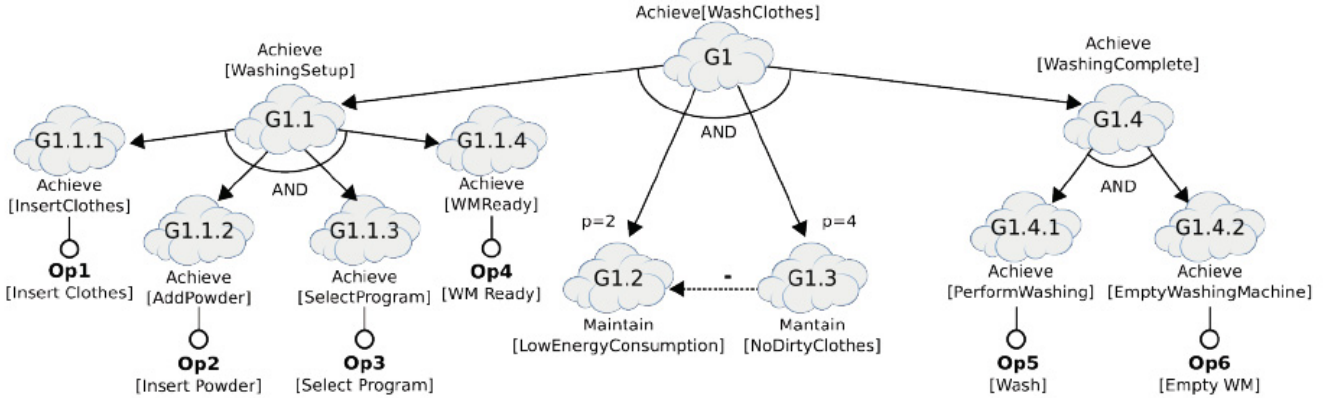
### 2. 관련 연구

#### 2.1 목표 모델

목표 모델은 소프트웨어 개발 프로세스 중, 요구사항 분석 단계에서 요구사항 도출에 이용되는 모델 중 하나이다[5]. 하나의 목표는 여러 목표들로 세분화 되어 계층 구조를 이룰 수 있다. 따라서 목표 모델은 일반적으로 트리 구조이며, 시스템의 '목표'를 나타내는 노드와 '목표'를 연결하는 엣지로 구성된다. 목표 모델에서는 트리의 부모 노드가 상위 목표를 의미하며, 자식 노드가 하위 목표를 의미한다. 대표적인 목표 모델로 KAOS, I\*, GRL, TROPOS 등이 있다[6]. 그림 1은 세탁 시스템의 KAOS 목표 모델이다[3].

#### 2.2 목표 모델 기반의 자가 적응 시스템

[2]와 [3]은 기존의 KAOS 목표 모델에 퍼지 개념을 도입하여 확장하거나[2] KAOS를 일반화 한 Fuzzy Live



(그림 1) 세탁 시스템의 KAOS 목표 모델[3]

Adaptive Goals for Self-adaptive systems (FLASGS)[3]를 제안하였다. 또한 목표(goal)를 런 타임 개체(entity)로 변형하고 적응 전략 (adaptation strategy)을 표현한 새로운 목표 유형인 adaptation goal을 제안하였다. 목표 만족도는 설계 시점에 각 목표들마다 도메인 전문가에 의해 정의된 멤버십 함수를 통해 평가된다.

### 2.3 목표 모델에서의 가중치

가중치는 목표 트리에서 사용자의 결정에 고려되는 여러 속성들의 영향력을 수치로 표현한 데이터이다. 가중치는 0과 1사이의 실수 값이고 가중치의 총 합은 1이다. 앞의 두 제약 조건으로 정규화 과정을 생략할 수 있다[7].

가중치와 제약 조건을 수식으로 정리하면 다음과 같다.

$$w_i \in R$$

$$\forall i = [1..n], 0 \leq w_i \leq 1$$

$$\sum_{i=1}^n w_i = 1$$

[8]은 정책과 환경간의 관련 정도를 가중치로 나타내고 시스템이 외부 환경에 대해 수집한 데이터나 사용자로부터의 피드백을 통해 비슷한 환경에 따라 환경과 사용자 기호에 적절한 정책이 실행되도록 정책 가중치를 최적화하는 방법을 제안하였다. 사용자 피드백에 따라 가중치를 변화 시키는 방법으로 최적화 방법 중 하나인 Sequential Quadratic Programming (SQP)method를 이용하였다.

[9]는 예상하지 못한 문제들을 식별하는 능력을 향상시킬 수 있도록 해주는 방법론인 goal-based contextual problem detection and management(GCPDM) method를 제안하였다. 이 논문에서는 목표 모델에 고정된 가중치를 부여하였다. 설계 시점에 부여된 고정된 가중치를 통해 목표 모델의 평가가 이루어지기 때문에 환경이 변화 될 경우 부적절한 평가를 보일 수 있다.

## 3. 가중치 획득과 변경

본 장에서는 여러 하위 목표들로 구성된 상위 목표를 표현하는 방법으로 목표 공간을 정의하고 시스템 설계 시

점에 가중치를 획득하는 방법과 설계 시점에 획득된 목표 모델의 가중치를 시스템 동작 환경에 따라 런 타임에 동적으로 변경하는 방법을 제안한다.

목표 모델에서의 가중치는 하위 목표들 간에 상위 목표를 달성하는데 미치는 영향력을 나타내며 설계 시점에 도메인 전문가에 의해 부여되고 사용자의 요구도 함께 반영된다.

### 3.1 목표 공간 (Goal Space)

본 논문에서는 d개의 하위 목표로 세분화된 상위 목표를 하위 목표들로 구성된 d차원의 유클리드 공간으로 추상화 하고 추상화된 상위 목표는 각 옛지에 부여된 가중치의 벡터로 표현한다.

목표 공간을 수식으로 정리하면 다음과 같다.

$$\text{목표 공간 } G^d \equiv \mathbb{R}^d$$

$$G^d = \langle w_1, \dots, w_d \rangle$$

d는 공간의 차원(dimension)으로 하위 목표의 개수와 같다. w는 하위 목표에 부여된 가중치 값 이다.

### 3.2 가중치 획득

Value Tree Analysis (VTA)는 일반적으로 의사 결정과 의사 결정에 영향을 미치는 속성들 사이의 분석에 이용되는 기법으로 시스템을 설계하는 시점에 도메인 전문가가 상위 목표와 하위 목표 사이의 가중치를 VTA의 Preference Elicitation을 이용하여 획득한다[7].

그러나 설계 시점에 획득 된 가중치가 런 타임 환경에 적절하지 않을 수 있기 때문에 테스트 시점이나 런 타임에 동적으로 변경 될 수 있어야 한다. 따라서 시스템은 시스템 테스트 시점이나 런 타임에 가중치 변경 방법의 실행 조건을 만족하는 상황에 가중치를 동적으로 변경한다.

### 3.3 가중치 변경

가중치 변경 시에는 Sequential Quadratic Programming (SQP) method를 이용하여 자동으로 가중치를 획득한다[8]. SQP method는 비선형 프로그래밍 문제(non-linear programming problem)를 반복법 (iterative method)을 통해 해를 구하는 방법으로 비선형 프로그래밍

문제는 해를 구하는 목적 함수(objective function)와 해에 대한 제약 조건(constraints)으로 구성된다[10].

본 논문에서는 Matlab의 최적화 틀에서 제공하는 *fmincon* 함수를 이용하여 새로운 가중치를 획득한다. 이 함수는 제약이 있는 스칼라 함수의 최솟값을 계산하는 함수이다[11].

가중치 변경 방법이 실행 될 조건은 다음과 같다.

1. 시스템의 상태는 '정상'이다.
  2. 목표 평가 결과가 임계값을 넘지 못하였다.
  3. 목표 평가 결과를 향상시킬 정책은 모두 수행되었다.
- 조건 1과 2는 시스템 외부 환경의 변화를 의미하며, 조건 3은 시스템의 자가 적응과 관련된 조건이다.

d차원 유클리드 공간에서 설계 시점에 설정된 가중치  $w$ 와 환경 변화가 반영된 새로운 가중치  $w'$ 을 서로 다른 두 점을 가리키는 벡터로 간주하고, 공간에서 두 점 사이의 거리를 구하는 공식을 이용하여  $w'$ 을 획득한다.  $w$ 는 사용자의 요구가 반영된 가중치이기 때문에  $w$ 와  $w'$ 의 거리(가중치 변화)는 최소가 되어야 한다.

가중치에 대한 목적 함수를 표현하면 다음과 같다.

$$\|w - w'\| \equiv \sqrt{\sum_{i=1}^n ((w_i - w'_i)^2)}$$

위의 목적 함수를 비선형 최적화 문제로 정리하면 다음과 같다.

$$\min \|w - w'\|$$

목적함수에 대한 제약조건은 다음과 같다.

1. 임계값  $\leq \sum_{i=1}^n w'_i * a(g_i)$
2.  $\sum_{i=1}^n w'_i = 1$
3.  $0 \leq w'_i \leq 1 (i = [1, n])$
4.  $w_i$ 사이의 순위는  $w'_i$ 에서도 유지된다

조건 1은 환경이 반영된 적절한 가중치를 획득하기 위한 조건이다. 조건 2와 3은 2.3절에 기술된 가중치에 대한 조건이고 조건 4는 설계 시점에 부여된 가중치에 반영된 사용자의 요구와 관련된 조건이다.

### 3.4 목표 만족도 (Goal Achievement)

목표 만족도는 하위 목표의 만족도와 가중치의 곱으로 구한다.

목표 만족도를 구하는 식은 다음과 같다.

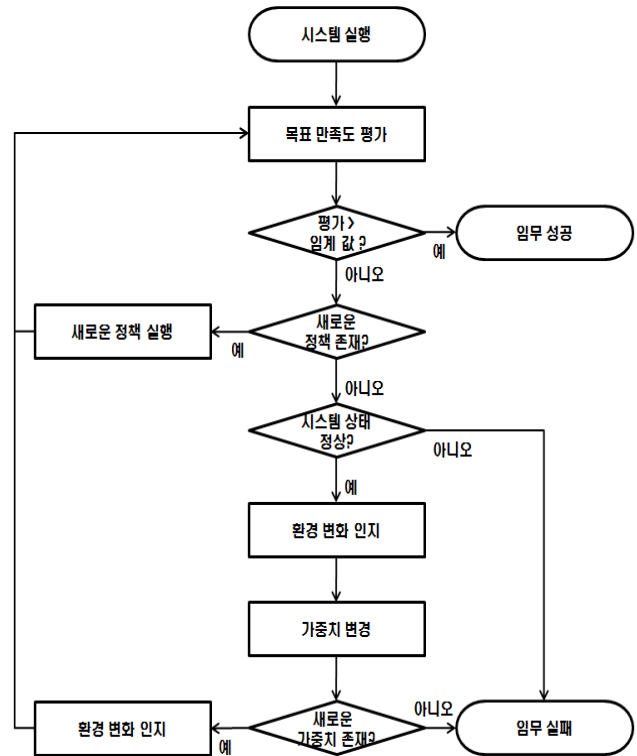
$$a(G) \equiv \sum_{i=1}^n w_i * a(g_i) \quad (i = 1, \dots, n)$$

목표 만족도  $a(g_i) \in [0, 1]$   
 가중치  $w_i \in [0, 1]$

G는 상위 목표, g는 G의 하위 목표이고 a(g)는 하위 목표의 만족도이다.

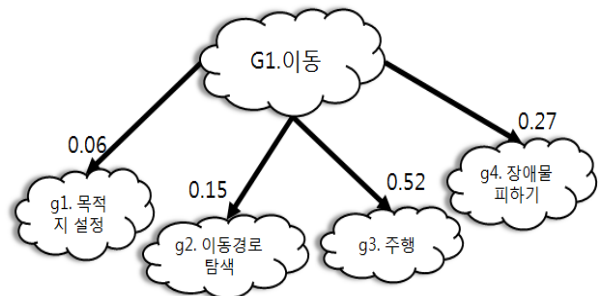
그림 2는 시스템 자가 적응(self-adaptation) 흐름도이다.

## 4. 적용 사례 : 공간 감시 로봇



(그림 2) 시스템 자가 적응(self-adaptation)

흐름도



(그림 3) '이동' 목표 모델

제안 방법의 유효성 평가를 위해 본 절에서는 가정에서의 공간 감시 로봇에 적용하여 사례 연구에 대한 시뮬레이션 실험을 하였다. 본 장의 공간 감시 로봇은 사용자 대신 사용자가 지정한 장소로 이동하여 그곳의 상황을 카메라로 찍어 사용자에게 영상을 전송해 주는 임무를 수행한다. 실험에서 '사용자가 입력한 목적지에 5분 안에 도착한다.'라는 공통적인 임무를 로봇에게 부여하고 로봇을 일반 노면(사례 1)과 카펫이 깔린 노면(사례 2)을 주행하는 시뮬레이션을 통해 실행 환경 변화에 대해 변경된 목표 가중치가 적절한 평가 결과를 보이는지를 검증하였다.

그림 3은 '이동'이라는 목표의 목표 모델이다.

### 4.1 실험 초기 데이터

표 1은 설계 시점에 일반 노면을 기준으로 획득된 가중치이다. VTA의 Preference Elicitation의 가중치 획득 방법 중 상위 목표에 대한 하위 목표의 영향력 순위를 기

<표 1> 초기 가중치

w1	w2	w3	w4
0.06	0.15	0.52	0.27

<표 2> 사례 1에 대한 로봇 임무 수행 결과

상태	g1	g2	g3	g4	a(이동)	임계값
임무 수행 후	1	0.9	1	1	0.985	0.85

<표 3> 사례 2에 대한 로봇 임무 수행 결과

상 태	g1	g2	g3	g4	a(이동)	임계값
①	1	0.9	0.7	1	0.777	0.85
②	1	0.9	0.72	1	0.839	0.85
③	1	0.9	0.72	1	0.850	0.85

<표 4> 변경 된 가중치

상태	w1	w2	w3	w4
변경 전	0.06	0.15	0.52	0.27
변경 후	0.0793	0.149	0.4824	0.2893

반으로 효율적이고 직관적으로 획득하는 방법인 Rank-ordered centroid (ROC) formula를 이용하여 가중치를 획득하였다[12].

#### 4.2 실험 결과

일반 노면과 카펫이 깔린 노면(일반 노면보다 로봇의 바퀴와 노면 사이의 마찰력이 증가한 환경)에 대한 시뮬레이션을 통해 목표 성취도를 측정하고 결과에 따라 결과 데이터이다.

##### 사례1. 일반 노면

표 2는 일반 노면에서의 로봇 임무 수행 결과 데이터이다. 시스템 자가 적응 흐름(그림 2)에 따라 '이동'목표 만족도가 임계값 이상이므로 시스템은 성공적으로 '이동' 임무를 수행하였다고 평가 되었다.

##### 사례2. 카펫이 깔린 노면

표 3은 카펫이 깔린 노면에서의 로봇 임무 수행 결과 데이터이다. 상태①은 카펫이 깔린 노면에서 첫 실행 결과 데이터이다. 시스템 자가 적응 흐름에 따라 '이동'목표 만족도가 임계값 미만이므로 로봇의 '이동'임무 수행이 실패로 되고 자기 적응을 위한 정책이 수행된다. 상태②는 로봇이 새로운 정책을 찾아 임무를 수행한 결과 데이터이다. 새로운 정책으로 로봇의 성능은 이전보다 조금 향상 되었으나 '이동'목표 만족도가 임계값 미만이므로 로봇은 '이동'임무 수행을 실패하였다고 평가되었다. 3.2절에서 제안된 가중치 변경 방법이 실행 될 조건에 따라 로봇은 가중치 동적 변경 방법을 통해 새로운 가중치를 획득하였다. 상태③은 가중치 변경 후 '이동'임무 수행 결과 데이터이다. 변화된 환경이 반영된 가중치를 통해 로봇의 '이동'임무 수행에 대한 적절한 평가가 이루어졌다.

표 4는 3.3절에서 제안한 가중치 동적 변경 방법으로

획득한 가중치이다.

## 6. 결론

본 논문에서는 환경 변화에 따라 가중치를 동적으로 변경하여 변화된 환경이 반영된 평가를 가능하게 하는 방법을 제안하였고 공간 감시 로봇에 적용하여 유효성을 평가하였다. 제안한 방법을 통해 기존의 고정된 가중치 기법의 목표 평가 방법보다 다양한 환경에 대해 신뢰성 높은 평가가 가능하다.

## 참고문헌

- [1] P. Horn, "Autonomic computing: IBM perspective on the state of information technology", IBM T.J. Watson Labs, NY, 2001
- [2] Luciano Baresi, Liliana Pasquale, "Adaptive Goals for Self-Adaptive Service Compositions", 2010 IEEE International Conference on Web Services
- [3] Luciano Baresi, Liliana Pasquale, and Paola Spoletini, "Fuzzy Goal for requirements-driven Adaptation", 2010 18<sup>th</sup> IEEE International Requirements Engineering Conference
- [4] Shan Tang, Xin Peng, Yijun Yu, and Wenyun Zhao, "Goal-Directed Modeling of Self-adaptive Software Architecture", BPMDS 2009 and EMMSAD 2009, LNBIP 29, pp. 313 - 325, 2009
- [5] Axel van Lamsweerde, "From System Goals to Software Architecture", Formal Methods for Software Architectures, M. Bernardo & P. Inverardi (eds), LNCS 2804, Springer-Verlag, 2003.
- [6] Gil Regev, Alain Wegmann, "Where do Goals Come from: the Underlying Principles of Goal-Oriented Requirements Engineering", 13th IEEE International Conference on Requirements Engineering
- [7] MCDA Team, "Value Tree Analysis Theory", [www.mcda.hut.fi/value\\_tree/theory/theory.pdf](http://www.mcda.hut.fi/value_tree/theory/theory.pdf)
- [8] Konstantinos Kakousis, Nearchos Paspallis, and George A. Papadopoulos, "Optimizing the Utility Function-Based Self-adaptive Behavior of Context-Aware System Using User Feedback", OTM 2008, Part I, LNCS 5331
- [9] Jae Sun Kim, Sooyong Park and Vijayan Sugumaran, "Contextual problem detection and management during software execution in complex environments", Industrial Management & Data Systems, Vol. 106 No. 4, 2006 pp. 540-561
- [10] Paul T. Boggs, Jon W. Tolle, "Sequential Quadratic Programming", Acta Numerica (1996), vol4. pp.1-51
- [11] Chong, E.K.P., Zak, S.H., "An Introduction to Optimization, 2nd edn", John Wiley & Sons Pvt. Ltd, Chichester (August 2001)
- [12] F.Hutton Barron, Bruce.E.Barrett, "Decision Quality Using Ranked Attribute Weights", Management Science, Vol.42, No.11, November 1996