

# 분산 환경에서의 번역 시스템의 개발 \*

: 사전 구축과 테스트 환경을 중심으로

이현아 장병규 강인호 이신목<sup>○</sup> 김길창  
한국과학기술원 전자전산학과

{halee,bgjang,ihkang,smlee}@csone.kaist.ac.kr, gckim@cs.kaist.ac.kr

Development of an MT System on Distributed Environment  
: about lexicon construction and test environment

Hyun Ah Lee Byung Gyu Jang In Ho Kang Shin Mok Lee<sup>○</sup> Gil Chang Kim  
Dept. of Electrical Engineering & Computer Science, KAIST

## 요 약

본 논문에서는 영한 기계번역 시스템 ALKOL의 개발 과정에서 사용된 분산 환경을 사전 구축과 테스트 환경을 중심으로 설명한다. 번역 시스템 개발은 시스템 개발자, 사전 개발자, 검사자(tester)의 세 요소를 중심으로 이루어지는 작업으로, 효율적인 개발을 위해서는 개별 요소의 작업이 독립적으로 진행되면서 유기적으로 연결될 수 있는 환경이 마련되어야 한다. ALKOL에서는 이를 위하여 번역 시스템을 번역 서버, 사전 서버, 사전 편집기, 사용자 인터페이스의 클라이언트/서버 분산 환경으로 구성하고 검사자를 위한 테스트 환경을 제공하여 효율적인 번역 시스템 개발 환경을 구축한다. 번역 시스템을 분산 환경에서 개발하여 시스템 개발자, 사전 개발자, 검사자가 각자의 작업을 병렬적으로 진행할 수 있고 실시간으로 갱신되는 시스템과 사전 정보를 이용하여 작업을 진행할 수 있어 번역 시스템 개발의 효율성을 높일 수 있다.

## 1 서론

일반적으로 기계번역 시스템은 번역 사전으로 대표되는 번역 지식을 이용하여 원시 언어의 문장을 목적 언어의 문장으로 번역한다. 사용자의 요구에 부합하는 실용성 있는 번역 시스템을 얻기 위해서는, 다양한 문장에 대한 테스트를 통하여 시스템과 사전에 대한 지속적인 튜닝(tuning) 과정을 거쳐야 한다. 테스트를 통한 번역 시스템과 번역 사전의 튜닝은 기계번역 시스템의 개발에서 가장 중요한 과정인 동시에 가장 많은 시간과 노력이 필요한 부분이다.

기계 번역 시스템의 테스트와 튜닝은 여러 명의 검사자(teacher)가 번역 시스템의 결과를 분석하고, 분석 과정에서 발견된 문제를 시스템 개발자와 사전 개발자에게 전달하여, 개발자들이 시스템의 성능을 향상시키고 사전 내용을 보장하는 과정으로 이루어진다. 이처럼 기계번역 시스템의 개발, 테스트, 튜닝 작업은 시스템 개발자, 사

전 개발자, 검사자가 상호 작용하는 형태가 된다.

본 논문에서는 영한 기계번역 시스템 ALKOL의 개발에 사용된 분산 환경을 소개한다. ALKOL에서는 번역 시스템을 번역 서버와 사전 서버로 구성하고, 일반 사용자를 위한 사용자 인터페이스와 사전 개발자를 위한 사전 편집기를 클라이언트 형태로 제공한다. 여기에 시스템 검사자가 테스트 작업을 원활히 할 수 있도록 테스트 환경을 제공하여, 시스템 개발자, 사전 개발자, 검사자가 효율적으로 작업할 수 있는 환경을 제공한다.

본 논문은 다음과 같이 구성된다. 2절에서는 영한 기계번역 시스템 ALKOL을 간단하게 소개하고, 3절에서는 ALKOL의 분산 개발 환경을 설명한다. 4절에서는 사전 구축 환경을 소개하고, 5절에서는 ALKOL의 테스트와 튜닝 지원 환경에 대하여 설명한다. 6절에서는 결론을 맺는다.

\* 본 연구는 공군본부 연구과제 "군사정보 기계번역 시스템 용역 개발"에 의해 지원되었음.

## 2 영한 기계번역 시스템 ALKOL

ALKOL은 공군의 정보 분야 영어 문서의 번역을 위해 개발된 영한 번역 시스템이다[1].

ALKOL은 사전에 저장된 어휘화된 규칙을 이용하는 변환 방식의 기계번역 시스템이다. 번역은 형태소 분석, 품사 태깅, 분석 전처리, 구문 분석, 변환, 생성의 단계로 수행되고 번역 사전은 어휘-구문-변환-생성의 네 단계가 연결된 형태로 구성된다. 구문 사전은 구문 분석을 위한 어휘화된 분석 규칙을 가지고, 규칙에 포함된 어휘의 어휘 사전 엔트리에 연결된다. 변환 사전은 어휘 사전이나 구문 사전에 연결되고 번역어 선택을 위한 정보와 구문 사전의 규칙으로 분석된 영어 구조를 한국어 구조로 변환하기 위한 정보를 가진다. 생성 사전은 어휘 사전이나 변환 사건의 엔트리를 한국어로 만들기 위한 구문 정보와 형태소 정보를 가진다. 각 사건의 엔트리들은 고유한 번호(entry ID)로 구분된다. 예를 들어 동사 attack은 어휘 사전에 attack1로, 명사 attack은 attack2로 지정되고, attack이 포함된 구문 분석 규칙을 가지는 구문 사전 엔트리는 attack1.1, attack2.on1 등으로, 이에 연결된 변환 사전 엔트리는 attack1.1.1, attack2.on1.1 등으로 표현된다. 사전은 하나의 시스템 사전을 포함하고 하나 이상의 사용자 사전이 추가될 수 있으며, 각 사전은 어휘-구문-변환-생성 네 단계의 동일한 구조로 구성된다.

## 3 분산 환경에서의 번역 시스템 개발

일반적으로 기계번역 시스템의 개발은 번역 프로그램을 개발하는 시스템 개발자와 사전 구축을 담당하는 사전 개발자(또는 사전 구축자)와 시스템에 대한 테스트를 담당하는 검사자(tester)를 중심으로 이루어진다. 이 때, 시스템 개발자와 사전 개발자, 검사자가 서로 유기적으로 연결된 환경에서 독립적으로 작업을 수행할 수 없다면 번역 시스템 개발의 효율성이 저하될 수 있다. 예를 들어, 여러 명의 검사자가 각각 다른 사전과 시스템을 이용하여 테스트를 수행한다면 올바른 평가 결과를 얻을 수 없다. 동일한 사전과 시스템을 이용하더라도 가장 최근에 갱신된 사전과 시스템을 실시간으로 바로 이용할 수 없다면 사전과 시스템이 업데이트 되기를 기다려야 한다. 따라서, 테스트 작업 뿐만 아니라 시스템 개발과 튜닝 작업 등의 개발 과정의 효율성이 크게 저하된다.

ALKOL에서는 이와 같이 동시 다발적으로 진행되는 사전 개발과 시스템 구축, 시스템 테스트와 튜닝을 실시간으로 처리할 수 있도록 분산 환경을 구축한다. 그림 1은 ALKOL의 개발 환경을 보인다.

개발 환경은 크게 사전 서버, 번역 서버, 사용자 인터페이스, 사전 편집기, 테스트 지원 틀로 구성된다.

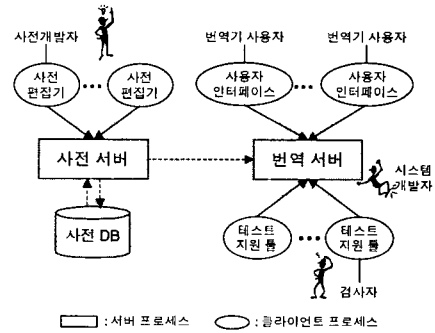


그림 1: ALKOL의 개발 환경

번역 서버는 사전 서버에서 사전 정보를 얻어 번역을 수행한다. 일반 사용자는 사용자 인터페이스를 통해 영어 문장을 번역 서버로 보내고, 번역 서버에서는 사전 정보를 이용하여 입력 문장에 대한 번역을 수행하고 결과를 사용자 인터페이스로 보낸다. 사용자 인터페이스를 번역 서버에 대한 클라이언트라고 보면, 사전 서버에 대한 클라이언트로는 사전 편집기가 존재한다. 사전 편집기는 사전 개발자가 새로운 사전 정보를 입력하거나, 기존의 사전 정보를 검색하거나 수정하거나 삭제하는 등의 작업을 편리하게 할 수 있도록 환경을 제공한다. 사전 서버에서는 사전 편집기에서 들어오는 요청을 받아서 데이터베이스에 저장된 사전 정보에 대한 입력, 검색, 수정, 삭제 작업을 수행한다. 사전 서버의 입장에서는 번역 서버도 하나의 클라이언트 프로세스가 된다. 번역 서버에서는 사전 서버에 사전 정보를 요청하여 번역을 수행한다. 테스트 지원 틀은 검사자가 시스템에 대한 평가를 수행하고 번역 과정에서 발생한 문제를 사전 개발자와 시스템 개발자에 전달할 수 있는 환경을 제공한다.

이와 같이 ALKOL에서는 번역 시스템의 개발 환경을 분산 환경으로 구축하여 시스템 개발자와 사전 개발자, 검사자의 작업의 효율성을 높일 수 있다. 그림 1에서 볼 수 있듯이, 분산 환경에서는 다중 사용자의 동시 작업이 가능하고 각 작업은 바로 시스템에 반영된다. 번역 서버와 사전 서버는 서로 독립적으로 동작하면서 시스템 개발자와 사전 개발자 각각에 의해 항상 최근에 갱신된 상태를 유지한다. 형태소 분석에서 생성까지의 번역 서버 단계의 개발자는 담당 프로그램에 대한 개발을 독립적으로 수행하여 최신 프로그램을 서버에 반영할 수 있다. 여러 명의 사전 개발자는 사전 편집기를 통해 각자 사전 정보에 접근할 수 있고, 사전 서버는 최신의 사전 정보를 클라이언트에 제공한다. 검사자는 항상 최근에 갱신된 시스템과 사전에 의한 번역 결과를 테스트 지원 환경을 이용하여 평가하고, 평가 결과를 시스템 개발자와 사전 개발자에게 전달한다.

따라서, 시스템 개발자와 사전 개발자, 검사자는 각자 독립적인 환경에서 병렬적으로 작업을 수행하면서 실시간으로 업데이트된 시스템, 사전 정보, 평가 결과를 이용할 수 있다. 번역 서버와 사용자 인터페이스, 사전 서버와 번역 서버, 사전 편집기와 사전 서버는 TCP/IP를 이용하여 연결된다[5, 6].

다음 절에서는 사전 구축 환경과 테스트 환경을 소개한다. 사전 구축 환경에서는 사전 서버의 구조와 사전 편집기에 대해서 소개한다. 테스트 환경에서는 ALKOL 개발 과정에서 사용한 테스트 지원 툴에 대해서 소개한다. 기계번역 시스템 개발의 최종 단계에서는 다양한 입력 문장에 대한 테스트와, 그 결과를 이용한 시스템과 사전의 지속적인 튜닝이 필요하다. 테스트 지원 툴에서는 검사자가 효율적으로 시스템을 테스트할 환경을 제공하는 동시에 검사 결과를 시스템 개발자와 사전 개발자에게 자동으로 전달하는 환경을 제공한다.

#### 4 사전 구축 환경

본 절에서는 ALKOL의 사전 구축 환경을 소개한다. ALKOL의 사전은 어휘-구문-변환-생성의 네 단계로 구성되며, 하나의 시스템 사전과 하나 이상의 사용자 사전을 가질 수 있다. 번역 사전은 데이터베이스로 저장되고, 번역 서버와 사전 편집기는 사전 서버에 접속하여 저장된 사전 정보를 사용할 수 있다. 사전 서버는 하나 이상의 요청을 처리할 수 있고 번역 서버와 사전 편집기는 항상 최신의 사전 내용을 이용할 수 있으므로, 번역기 개발과 사전 구축 작업이 효율적으로 진행될 수 있다.

다음에서는 사전 서버의 구조와 사전 편집기에 대해서 설명한다. 사전 서버에서는 사전 내용에 대한 접근을 처리하고, 사용자 확인 과정 등을 통해 사전 정보에 대한 접근 권한을 제어한다. 사전 편집기에서는 사용자가 사전 정보를 쉽게 구축할 수 있는 환경을 제공한다.

##### 4.1 사전 서버의 구조

그림 2는 사전 서버의 구조를 보인다. ALKOL은 하나의 시스템 사전과 하나 이상의 사용자 사전을 가지며, 각각의 사전은 별도의 데이터베이스에 저장된다. 데이터베이스는 GNU에서 제공하는 gdbm을 이용하여 구축한다[7]. 사전에 접근하고자 하는 클라이언트들은 사전 영역을 지정하여 사전 서버에 연결된다.

사전 서버에는 사전 편집기와 번역 서버의 두가지 클라이언트가 연결된다. 번역 서버에서는 필요로 하는 사전 영역과 사전 엔트리의 번호를 사전 서버에 보내고 사전 서버에서는 해당되는 사전 내용을 번역 서버로 전달한다. 번역 서버는 읽기 전용으로 사전 서버에 접속한다.

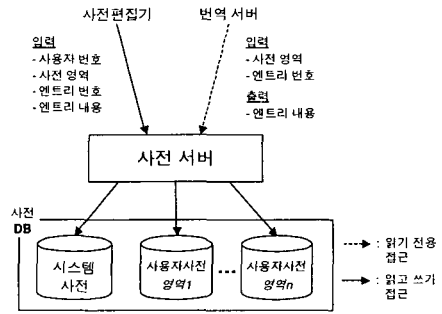


그림 2: ALKOL의 사전 서버

이에 반하여 사전 편집기는 읽기 쓰기 접근으로 사전 서버에 접속한다. 이 때 사전 서버는 사전 내용에 대한 접근 권한을 제한하기 위해서 사전 개발자의 사용자 번호(user ID)를 확인한다. 서버에 등록된 사용자는 사전 내용을 입력/검색/수정/삭제할 권한을 가질 수 있지만, 등록되지 않은 사용자는 검색만 할 수 있다. 사전 편집기는 사전 서버에 접속할 때 사용자 확인과 함께 편집할 사전 영역을 지정한다. 사전 개발자는 사전 편집기 상에서 지정한 사전 영역의 사전 정보에 대한 입력/검색/수정/삭제 작업을 수행한다. 사전 서버에서는 각 사용자의 작업 내용을 추적할 수 있게, 각 사용자의 접속 상황과 사전 변경 작업 내용을 로그 파일(log file)로 생성한다.

##### 4.2 사전 편집기

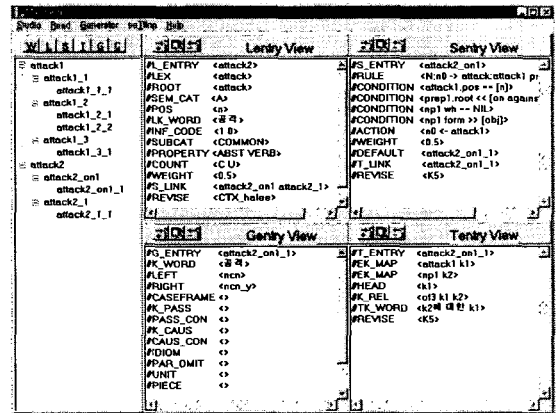


그림 3: ALKOL의 사전 편집기

그림 3은 ALKOL의 사전 편집기를 보인다. 사전 편집기는 크게 다섯개의 창으로 구성된다. 우측의 네 개의 창은 어휘-분석-변환-생성의 네 단계 사전의 내용을 보인다. 각 창에서는 새로운 사전 정보를 입력하거나, 기존의 사전 정보를 수정하거나 삭제할 수 있다. 네 단계의 사전 엔트리들은 연결된 형태로 구성된다. 사전 편

집기의 좌측 창은 어휘-분석-변환-생성의 연결된 사전 구조를 다단계 형식으로 나타내어, 사전 구조를 쉽게 파악할 수 있다. 그림 3에서는 'attack'에 대한 attack2 → attack2\_on1 → attack2\_on1.1에 대한 사전 내용을 보인다. 이 때 만일 attack2를 사용자가 삭제하면 사전 편집기에서는 attack2에 연결된 분석-변환-생성 사전 엔트리도 동시에 삭제할 수 있는 옵션을 제공한다. 또한 사전 내용을 새롭게 입력할 때 값들을 자동으로 구성해 줄 수 있으며, 입력하거나 수정하는 사전 내용이 문법에 맞게 작성되었는지 검사하는 기능도 제공한다.

사전 편집기는 프로그램이 시작될 때 사용자로부터 사용자 번호와 편집하고자 하는 사전 영역을 입력받아 사전 서버에 접속한다. 사전 서버는 지정한 사용자 번호와 사전 영역을 가지고 사전 편집기의 접근 권한을 결정한다. 사전 서버와 마찬가지로 사전 편집기도 로그 파일을 유지하여 사용자가 자신의 사전 작업 내용을 추적할 수 있게 돕는다.

## 5 테스트와 튜닝 환경

본 절에서는 ALKOL의 테스트와 튜닝 환경에 대해서 소개한다. 그림 4는 ALKOL의 테스트 환경을 도식화하여 보인다. 테스트 환경은 크게 테스트 지원 툴과 분석결과 전달기로 구성된다. 그림에서와 같이 검사자(tester)들은 테스트 지원 툴을 이용하여 번역 결과를 평가하고, 검사자들의 평가는 테스트 결과 데이터베이스(test DB)에 저장된다. 분석 결과 전달기는 데이터베이스에 저장된 오류 분석 결과를 분류하여 정기적으로 사전 개발자나 시스템 개발자에게 통보하여, 시스템과 사전에 대한 튜닝이 이루어진다.

다음에서는 검사자를 위한 테스트 지원 툴에 대해서 설명하고, 테스트 결과를 이용한 튜닝 과정에 대해서 설명한다.

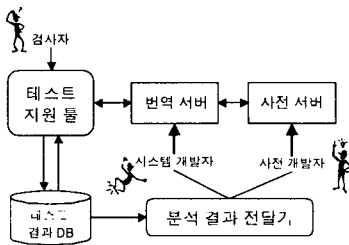


그림 4: ALKOL의 테스트 환경

## 5.1 테스트 환경

기계번역 시스템의 테스트는 다양한 입력 문장에 대하여 시스템이 올바른 결과를 출력하는지, 그렇지 않다면 어느 부분에 문제가 있는지를 파악하는 작업이다. 테스트 작업은 문장 단위로 이루어진다.

ALKOL에서는 대상 영역에서 수집된 각 문장에 고유한 번호를 부여하고 해당 문장에 대해서 번역 결과를 평가하기 위한 테스트 지원 툴을 개발하여, 테스트와 튜닝 작업을 지원한다. 그림 5는 테스트 지원 툴을 보인다. 테스트 지원 툴은 텍스트 기반 인터페이스로 구성된다.

```

Input:J33 [1]
J33 MA TG CK PS SA TR GR
error message : [GR] 생성 결과 어순에 문제가 발생
current status : MA[1], TG[1], CK[1], PSA[1], SA[1], TR[1], GR[0]
last update : Sat Aug 26 23:55:48 2000

input sentence : Some sections of the military are more severely affected than others.
ok output :
  군대의 약간의 부분은 다른 것보다 더 심하게 영향을 받습니다.

current output : 더 심하게 군대의 약간의 부분은 다른 것보다 영향을 받습니다.

[e]. error message [c]: current status [a]: ALKOL.pl [d]: diff result
[n/m]: MA [1/1]: TG [x/x/1]: Chunk [f/f]: segment [s/S/1]: SA [1]: class
[k/K/s]: TR [g/G]: GR [a]: apply [n]: Any.sh [./:]: next/prev [q/Q]: quit
-----
K

[Korean Tree #1-1]
affected1 |--- agt-> section1 |--- of1-> military2
          |--- cnp-> others1 |--- mod-> some1_3_1 some1_3_2
          |--- amod-> severely1

<영향을 받> |--- agt-> <부분> |--- of1-> <군대>
           |--- cnp-> <다른 것> |--- mod-> <약간의./어떤>
           |--- amod-> <심하게>

[Korean Tree #1-2]
affected1_1_1 |--- anl-> section1 |--- of1-> military2
# [영어][한성][발성]
  
```

그림 5: 테스트 지원 툴

툴의 기능은 번역 실행 기능, 번역 결과 보기 기능, 결과 평가 기능, 평가 결과 보여주는 네가지로 나뉜다.

- 번역 실행 기능

테스트 지원 툴에서는 문장 번호를 입력하면 해당되는 영어 문장을 읽어들이어 번역을 실행시킨다. 형태소 분석(MA), 태깅(TG), 명사구 인식(CK), 구문 분석 전처리(PS), 구문 분석(SA), 변환(TR), 생성(GR) 등의 번역 중간 과정들 중에서 원하는 부분만을 따로 실행하는 기능도 제공한다. 번역을 실행하면 입력 문장과 번역 결과 문장, 그리고 번역 과정에서 생기는 중간 결과가 모두 파일로 저장된다. 툴에서는 검사자가 입력한 새로운 문장에 대한 번역을 실행할 수 있는 메뉴도 제공한다.

- 번역 결과 보기 기능

그림 5에서는 문장 번호 J33번에 대한 번역 결과 보기의 예를 보인다. 입력 문장과 결과 문장은 툴의 화면에 바로 제시되고, 중간 결과들은 메뉴를 선택하여 볼 수 있다. 중간 결과 보기에서는 결과 텍스트 파일을 직접 볼 수도 있고, 툴에서 제공하는 보기 기능을 이용하여 검사자가 이해하기 쉬운 구조로도 볼

수 있다 그림 5의 아래 부분에서는 변환 결과인 한국어 의존 구조를 트리 형식으로 나타난 형태를 보인다.

- 번역 결과 평가 기능

결과 평가에서는 번역 결과가 어느 단계까지 정확한 결과를 냈는지를 지정하고, 발견한 문제들에 대한 평가문을 작성할 수 있다. 그림 5의 **current status**는 현재 결과가 변환 단계(TR)까지는 정확하지만 생성 단계에서 문제가 발생했음을 0과 1로 표시한 예를 보인다. 검사자는 **error message**에 평가문을 남길 수 있다. 결과 평가가 끝나면 검사자는 현재 문장에 대한 평가를 완료했음을 선택하고, 틀에서는 현재 평가 결과, 즉 문제가 발생한 단계와 검사자의 평가문을 테스트 데이터베이스에 저장한다. 동시에 중간 결과 파일들을 별도의 공간에 저장한다. 저장된 중간 결과 파일들은 평가 결과 보여주기에서 결과 비교에 사용된다.

- 평가 결과 보여주기 기능

평가 결과 보여주기에서는 데이터베이스에 저장된 검사자의 평가 결과를 문장별로 출력한다. 평가 결과 보여주기에서는 결과 비교 기능도 제공한다. 개발자의 프로그램 수정과 같은 시스템 변동 사항이 있을 경우 이전에 올바르게 번역되는 문장들이 잘못 번역되는 경우가 발생할 수 있다. 평가 결과 보여주기에서는 평가 결과를 출력할 때, 결과 평가에서 저장했던 중간 결과 파일들과 현재 결과 파일들을 비교하는 기능을 제공한다. 예를 들어 그림 5에서는 기존에 올바르게 번역되던(OK output) 문장이 시스템 변경 이후 틀리게 번역된 경우를 보인다. 이때 화면 상단의 결과 비교란에서는 생성 단계의 결과가 기존의 결과와 다르다는 점을 MA TG CK PS SA TR \*GR의 형식으로 나타나게 된다. 이 정보를 이용하면 시스템 개발자가 시스템을 변경한 뒤 결과가 바뀐 경우들을 쉽게 찾을 수 있으므로 시스템 개발 과정의 문제를 줄일 수 있다.

테스트 지원 틀에서 제공하는 평가 결과 보여주기를 이용하여 분석 결과 전달기는 시스템 개발자와 사전 개발자에게 검사자가 분석한 오류를 자동으로 전달하여 튜닝 작업이 이루어지게 한다. 다음에서는 튜닝 환경에 대해서 설명한다.

## 5.2 튜닝 환경

테스트 지원 환경은 테스트 지원 틀과 분석 결과 전달기로 구성된다. 분석 결과 전달기는 테스트 지원 틀의 평가

결과 보여주기를 정기적으로 실행시켜, 현재까지 수행된 검사자의 평가 결과를 얻는다. 얻어진 평가 결과는 문제가 발생한 번역 단계와 검사자가 작성한 평가문을 문장별로 나열한 형태가 된다. 분석 결과 전달기는 평가 결과를 분류하여 문제가 발생한 번역 단계의 개발자에게 해당 문장 번호와 평가문을 전자메일을 통해서 자동으로 전달한다. 검사자는 미리 정의된 형식에 따라 평가문을 작성하므로, 전달기에서는 평가문의 형식을 해석하여 개발자에게 평가문을 전송할 수 있다.

평가 결과를 전달받은 시스템 개발자나 사전 개발자는 시스템과 사전 정보를 변경하게 되고, 변경된 사항은 구동중인 시스템에 실시간으로 반영된다. 따라서, 검사자와 시스템 사용자, 사전 개발자는 최신의 사전과 시스템을 이용하여 동시에 작업을 진행할 수 있으므로 시스템 개발과 튜닝 작업이 효율적으로 이루어질 수 있다.

## 6 결론

본 논문에서는 영한 기계번역 시스템 ALKOL 개발에 사용된 분산 환경을 보이고, ALKOL의 분산 환경이 효율적인 번역 시스템 개발 환경을 제공함을 보였다. 번역 시스템 개발 환경을 번역 서버와 사전 서버, 사전 편집기와 번역 인터페이스의 클라이언트로 구성하고 테스트 환경을 마련함으로써, 번역 시스템 개발에 효율성을 높이고 효과적인 튜닝 작업이 이루어질 수 있었다.

이와 같은 분산 환경은 지속적으로 사전 정보가 추가되거나 시스템이 변경되는 경우에는 개발 환경뿐만 아니라 실제 사용 환경으로도 이용할 수 있다. 만일 실제 사용 환경이 분산 환경을 지원하지 않는다면, 분산 환경을 통해 개발된 시스템을 실제 환경에 맞게 변환하는 작업이 필요하다. ALKOL의 경우에는 개인 사용자를 위하여 분산 환경에서 개발된 시스템을 개인용 컴퓨터(PC)의 Windows 환경으로 변환하는 작업이 진행될 예정이다. 또한, 사전과 테스트 결과 데이터베이스의 관리에서 발생할 수 있는 다중 사용자 처리 등의 문제점에 대한 고려가 필요하다[4].

## 참고문헌

- [1] 이현아, 임철수, 최명석, 강인호, 김길창, “공군 정보 영한 기계번역 시스템 : ALKOL,” 제 12회 한글 및 한국어 정보처리 학술대회, 2000.
- [2] 임철수, 이현아, 최명석, 장병규, 이공주, 김길창, “어휘화된 규칙에 기반한 영한 기계번역 시스템,” 한국정보과학회 추계학술대회 발표 논문집, 1997.

- [3] 한국과학기술원, “군사 정보 기계번역 시스템 개발 4차년도 최종 보고서,” 2000.
- [4] 장현숙, 임점미, 유원경, 홍기형, 박상규, “자동 번역용 대용량 번역 지식 DB 관리 시스템 설계 및 구현,” 한국정보과학회 춘계 학술대회 발표 논문집, 2000.
- [5] W. Richard Stevens, “TCP/IP Illustrated,” *Addison-Wesley*, 1998.
- [6] W. Richard Stevens, “UNIX network programming,” *Prentice Hall*, 1994.
- [7] Philip A. Nelson and Jason Downs, “GNU dbm : a Database Manager,” *Free Software Foundation Inc.*, 1999.