韓國 컴퓨터情報學會 論文誌
第13卷 第3號, 2008. 5.

2008-13-3-1-4

서비스 지향 구조 기반의 EST 서열 주해 시스템

남 성 혁*, 김 태 경**, 김 경 란***, 조 완 섭***

An EST Sequence Annotation System Based On Service Oriented Architecture

Seong-Hyeuk Nam*, Tae-Kyung Kim**, Kyoung-Ran Kim***, Wan-Sup Cho***

요 약

본 논문에서는 SOA 기반의 EST 서열 주해 시스템인 SeqWeB을 제안한다. SeqWeB은 EST 서열 주해에 사용되는 8개의 분석 프로그램 (Phrap, cross_match, RepeatMasker, ICAtools, TGICL, CAP 3, Phrap, BLAST)을 웹 서비스로 제작하고, BPEL (Business Process Execution Language)을 통해 8개의 서비스를 다양한 형태로 조합한다. BPEL로 조합한 서비스들은 표준 데이터 형식으로 통신하여 통합 시 상호 운용성을 보장한다. SeqWeB은 웹 서비스와 BPEL을 통한 약 결합 방식으로 통합하여, 기존의 애플리케이션 통합 방식보다 시스템의 확장과 수정이 쉬우며 유지보수 비용이 저렴하다. 또한, SeqWeB은 다른 서비스의 컴포넌트로 사용될 수도 있다. SeqWeB을 통해 SOA가 지향하는 재사용성 (Reusability)과 유연성 (Flexible)을 기반으로 기존과 다른 방식의 생물학 분야의 애플리케이션 통합방법론을 제시한다.

Abstract

In this paper, we present an EST sequence annotation system based on Service Oriented Architecture, called SeqWeB. We developed the web services of eight applications (Phred, cross_match, RepeatMasker, TGICL, ICAtools, CAP3, Phrap and Blast) which are located in sequence annotation process and integrated the web services through BPEL. SeqWeB uses an XML file format for data input and output to maximize interoperability between each application. SeqWeB can be extended or modified easily through some modification such as insertion, deletion and replacement because service-oriented architecture allows loose coupling between applications.

▸ Keyword : 서비스 지향 구조(Service Oriented Architecture), 웹 서비스(Web services), BPEL, EST 서열 주해(EST sequence annotation)

• 제1저자 : 남성혁 교신저자 : 조완섭
• 접수일 : 2008. 3. 20, 심사일 : 2008. 4. 13, 심사완료일 : 2008. 5. 24.
* 한국생명공학연구원/UST ** 충북대학교 정보산업공학과 *** 충북대학교 경영정보학과/BK21 U-Biz팀
※ 이 논문은 2006년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음.

Ⅰ. 서 론

EST(Expressed Sequence Tag)는 cDNA 서열 조각으로서[1], 저렴한 비용으로 신속한 대량 생산이 가능하여 유전자 발견 및 매핑에 널리 사용된다[2]. EST 서열로부터 유전자를 찾아내고 그 기능을 분석하는 EST 서열 주해는 생명 현상 규명을 위해 가장 기본적이고 필수적인 연구로서, 크로마토그램 (Chromatogram)에 대한 염기 호출 과정부터 시작하여 여러 단계의 전 처리 과정이 필요하다. 이 과정에서는 Phred[3], TGICL[4], BLAST[5] 등 다양한 분석 프로그램들이 사용된다.

다수의 분석 프로그램들을 활용한 기존의 EST 서열 주해 과정에는 여러 가지 한계가 있다. 첫째, EST 서열 주해의 특성상 다수의 분석 프로그램 간 연결 과정에서 입출력 데이터 형식 변환이나 데이터 파싱(parsing)을 위한 사용자의 별도 작업이 요구된다. 이 과정에서 데이터 소실, 오류 유입 등에 의해 주해 결과의 질이 저하될 수 있다. 둘째, 사용자는 다수의 분석 프로그램들을 활용하기 위해 서버 환경을 구축하고 해당 프로그램을 직접 설치, 운영해야 한다. 또한 서버 운영 및 분석 프로그램들에 대한 기본적인 학습이 필요하며, 분석 프로그램의 버전 업그레이드를 직접 관리해야 한다. 셋째, EST 서열 주해와 관련된 다양한 분석 프로그램들이 공개되어 있는데 이 중 기능은 같지만 플랫폼, 입출력 데이터 형식 등이 이질적인 경우가 많아 사용자는 서버에 설치된 분석 프로그램 이외의 다른 프로그램을 사용하기 위해 추가의 시간을 들여야만 한다.

EST 서열 주해의 중요성이 부각됨에 따라 기존 분석 프로그램들의 성능이 점점 향상되고 있으며, 새로운 분석 프로그램들이 지속적으로 개발되고 있다. 현 상황에서 확장이 용이하고, 사용자의 요구에 유연하고 민첩하게 대응할 수 있으며, 분석 프로그램 간 상호 운용성을 보장하는 SOA 기반의 통합 EST 서열 주해 시스템이 필요하다.

본 논문에서는 ⅰ) 웹 서비스와 BPM 기술을 기반으로 서비스 지향 아키텍쳐 (SOA) SW 구현 방법론을 제시하고 ⅱ) EST 서열 주해를 위해 분석 프로그램 간 상호 운용성을 보장하는 자동화된 웹 기반의 시스템인 SeqWeB을 제안한다.

SOA 기반의 SW 개발 방법론에서는 기존의 응용프로그램 및 API를 웹 서비스로 배포하는 과정을 제시한다. 또한 개발된 웹 서비스들을 코드 기반 또는 BPM 기술의 표준인 BPEL을 이용한 모델링을 통해 사용자가 원하는 기능들을 실시간으로 조합할 수 있는 프레임워크를 제시한다.

제안된 SW 개발 방법론을 기반으로 실제 EST 서열 주해 통합 시스템인 SeqWeB 구현한다. SeqWeB은 8개의 분석 프로그램 (Phred, cross_match, RepeatMasker, ICAtools, TGICL, CAP3, Phrap, BLAST)을 단위 웹 서비스로 배포하여 사용자가 원하는 주해 프로세스를 동적으로 구현할 수 있다. SeqWeB은 SOA가 지향하는 유연성(Flexibility), 민첩성(Agility), 통합(Integration)을 모두 구현하였으며, 사용자의 요구 및 외부 환경의 변화를 실시간으로 반영 가능한 실시간 분석 시스템 (Real-Time Analysis System)으로써 발전할 수 있다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련연구를 제시하고, 제3장에서는 서비스 배포 및 통합 방법론을 제시한다. 제4장에서는 SOA 개발방법론으로 구현한 SeqWeB 시스템을 설명하고, 제5장에서는 SeqWeB 시스템을 SW 관점 및 생명 정보학 통합 시스템 관점에서 비교 분석한다. 제6장에서 결론 및 향후 연구를 설명한다.

Ⅱ. 관련 연구

본 장에서는 EST 서열 주해과정과 이 과정을 자동화한 기존 EST 서열 주해 시스템에 대해 설명하고, 기존 시스템들의 한계점을 제시한다. 또한 서비스 지향 개발 방법론과 구현 기술에 대해 알아본다.

2.1 EST 서열과 주해 절차

EST(Expressed Sequence Tag)는 발현된 유전자인 cDNA의 단편으로, 서로 다른 생물종이나 세포, 기관에서 발현되는 유전자의 DNA 단편 서열들과 비교하여, 염색체 DNA로부터 유전자의 정확한 위치를 찾아내는데 사용된다[1,6]. EST 서열 주해는 특정 유전자 부위가 어떤 기능으로 발현되는가를 분석하는 것으로 DNA 칩 데이터 분석, SNP 분석, Motif 분석, Proteomics 등 여러 분야에 폭넓게 적용되며[2] 유전자 발견과 매핑을 위한 자원으로도 널리 사용되고 있다.

EST 서열 주해는 대상 종 (種)과 목적에 따라 다양한 전처리 과정을 거치나 일반적으로 그림 1과 같은 과정을 거쳐 이루어진다.

A단계는 생물학자가 실험을 통해 아직 기능이 밝혀지지 않은 유전자 서열을 얻는 과정이다. DNA 시퀀싱을 거친 후 결과물을 시퀀서로 처리하면 컴퓨터에 입력 가능한 크로마토그램 파일이 생산된다.

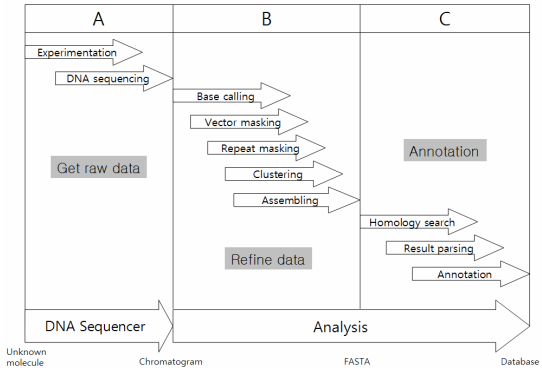

그림 1. EST 서열 주해 절차
Fig 1. EST sequence annotation procedure

B단계는 주해에 앞서 EST 서열을 정제하는 과정으로 다양한 분석 툴을 이용하여 5단계의 처리과정을 거친다. 염기 호출 (Base calling) 단계는 생물학자가 얻어낸 크로마토그램을 읽어 들여 각각의 해당 염기를 결정한다. 벡터 서열 제거 (Vector masking) 단계에서는 염기 호출 과정에서 읽은 염기 서열 중 cDNA 클로닝을 위해 사용한 벡터 서열이 전체 혹은 부분적으로 포함 될 수 있기 때문에 이를 제거한다. 반복 서열을 제거하는 과정(Repeat masking)은 DNA 서열 내에는 생물 종마다 유전자의 발현과 무관한 단순 반복되며 광범위하게 존재하는 염기배열을 제거한다. 벡터와 반복 서열 제거 과정은 EST 서열에 포함된 다양한 형태의 불필요한 부분을 제거하는 과정으로 양질의 서열을 얻기 위한 필수 과정이다.

다음은 확실하지 않은 대량의 DNA 서열 조각들을 클러스터링 (Clustering)하는 과정이다. 이 과정을 거치면서 기능별 클러스터의 대표 서열을 얻게 되고, 그 대표 서열을 통해 자신이 애초에 얻은 서열 전체의 기능을 예측하게 된다. 클러스터링 과정을 통해 얻게 된 대표 서열들은 어셈블링 과정을 거쳐 컨티그 파일로 생성된다.

C단계는 5단계의 정제 과정을 모두 마친 컨티그 서열을 이용하여 상동성 검색 (Homology search)을 수행하고 결과를 분석하여 기능을 예측하여 사용자에게 정보를 제공한다.

2.2 EST 서열 주해를 위한 관련 시스템

EST 서열 주해는 그림 1의 B, C 단계를 거치는 것이다. 이 과정은 최소 6단계(즉, 6개 프로그램)가 유기적으로 연결되어야만 한다. 초창기에는 각 분석 프로그램 간 연결을 위해 사용자의 수작업 또는 사용자가 직접 제작한 스크립트에 의해 처리하였다. 입출력 데이터 형식 변환, 서열 데이터 파싱 등을 위한 사용자의 수작업 때문에 시간이 오래 걸릴 뿐만 아니라 서열 데이터의 소실 또는 오류가 유입되어 잘못된 분석 결과의 원인이 되었다.

이러한 단점을 극복하고자 EST 분석 통합 시스템이 개발되었고, 대표적으로 ESTAP(EST Analysis Pipeline)[7], ESTAnnotator[8], GeneMaster[9] 등이 있다. 이 시스템들은 공개된 외부 데이터베이스를 이용하여 자동 또는 반자동으로 EST 서열을 정제하고 주해하는 서비스를 제공한다[10]. 이 분석시스템들은 공통적으로 여러 단계의 분석 과정을 자동화함으로써 처리 시간을 급격히 줄여주고 분석결과의 정확도를 높여준다. 하지만, 다음과 같은 특징 때문에 확장성과 유지 보수 측면에서 한계가 있다.

첫째, 이 시스템들은 로컬 시스템에 standalone 방식으로 설치되며, 프로그램과 데이터베이스의 최신 버전 유지를 위해 관리자의 업데이트가 필요하다. 둘째, 이 시스템들은 분석 단계마다 하나의 프로그램만으로 자동화 하여 다른 분석 프로그램에 대한 선택의 여지가 없다. 셋째, 이 시스템들은 강 결합(tight coupling) 방식의 통합으로 확장과 사용자의 요구사항에 신속히 대응할 수 없고 제시된 과정만을 거쳐야 한다. 마지막으로, 이러한 파이프라인 시스템은 웹 서비스로 제공되지 않으므로 다양한 응용에서 사용자가 필요한 기능을 사용할 수 없는 시·공간적 제약이 따른다. ESTpass[11] 시스템은 웹 기반으로 서비스를 제공하여 시·공간적 제약을 해결하였지만 내부적인 구조는 파이프라인을 이용한 시스템과 같다.

기존 시스템의 한계를 극복하려는 방안으로 서비스 지향 아키텍처 기반 웹 서비스 모델 및 BPM 아키텍처의 도입이 적합하다.

2.3 클러스터 컴퓨팅

서비스 지향 아키텍처는 표준 인터페이스와 메시지 프로토콜을 이용하여 전체 애플리케이션을 구축하는 소프트웨어 아키텍처이다[Gartner]. 그림 2는 SOA의 기본 구성 요소이다.

서비스 요청자 (Service requester)는 서비스 제공자(Service provider)에 의해 제공되는 하나 이상의 서비스를 이용하는 소비 주체이다. 서비스 제공자는 서비스 요청자가 호출 시 입력하는 값을 가공하여, 그에 해당하는 결과를 제공한다. 경우에 따라 서비스 제공자는 또 다른 서비스 제공자의 서비스를 이용하는 서비스 요청자가 될 수 있다. 서비스 레지스트리 (Service registry)는 서비스에 대한 정보(description)를 관리하며, 검색을 지원한다. 서비스 제공자는 자신이 제공하는 서비스를 등록하고, 서비스 요청자는 자

신이 원하는 서비스를 검색, 호출할 수 있다.

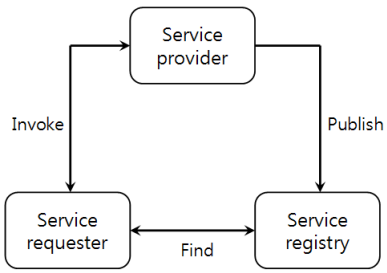

그림 2. 서비스 지향 구조의 구성
Fig 2. SOA Components

SOA는 웹 서비스 (Web Services) 기술 등을 이용하여 단위 응용 프로그램의 기능을 표준 인터페이스로 구현하고 서비스를 제공한다. 또한, XML 기반의 WSDL 문서로 외부에 공개하고, 사용자는 필요할 때 이 서비스를 UDDI에서 검색과 호출하여 원하는 애플리케이션을 조립하듯이 개발한다. 웹 서비스는 SOA를 실현하기 위한 기술이다. 웹 서비스는 SOAP, WSDL, UDDI 등의 표준기술을 기반으로 특정 플랫폼에 독립적이고 상호 운용성을 보장한다[12].

SOAP (Simple Object Access Protocol)은 Microsoft에서 제안하여 W3C의 표준 프로토콜로서, 웹 서비스의 요청 및 응답에서 사용되는 메시지 형식을 정의하고 있으며 방화벽 친화적 (firewall-friendly)이다. WSDL (Web Services Description Language)은 웹 서비스 이용에 필요한 인터페이스와 입출력 메시지의 형식을 기술하려고 이용된다. UDDI (Universal Description, Discovery and Integration)는 웹 서비스에 대한 디렉토리 서비스를 지원하려고 개발된 분산 레지스트리 표준으로 웹 서비스를 등록하고, 검색이나 바인딩하기 위한 메커니즘을 제공한다. XML (eXtensible Markup Language)은 SGML에서 파생된 단순하고, 매우 유연한 텍스트 형태의 언어이다[13]. Business Process Execution Language (BPEL)는 XML 기반으로 다수 서비스를 End-to-End 프로세스로 구성하는 언어이다[16]. BPEL은 웹 서비스, 자바 서비스, 사용자 정의 프로세스 등과 같이 이질적인 서비스들을 통합하기 위한 프로세스 설계와 조정 (Orchestration)의 기능을 제공한다.

III. 웹 서비스 제작 및 서비스 통합 기법

본 장에서는 웹 서비스의 배포 절차와 기법을 중심으로 설명한 후, 단위 웹 서비스들을 통합하기 위한 모델을 제시한다. [그림 3]은 웹 서비스의 제작과 배포 절차를 보여준다.

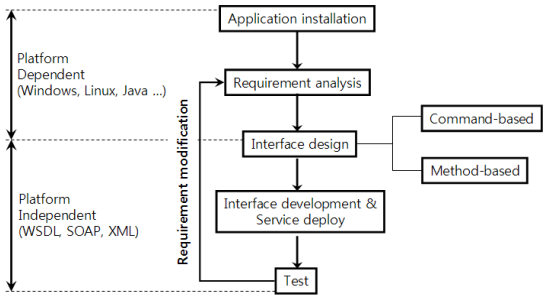

그림 3. 웹 서비스 제작 및 배포 절차
Fig 3. Design and deployment process of Web services

웹 서비스 대상인 애플리케이션은 요구 사항 분석 과정 이후 필요한 인터페이스를 설계한다. 인터페이스는 대상 애플리케이션을 호출할 때, 명령어 기반이나 메소드 기반으로 호출할 수 있으며, 설계된 인터페이스의 구현이 완료되면 이를 이용하여 웹 서비스로의 배포된다. 웹 서비스로 배포되면서 플랫폼 의존적이었던 기존의 애플리케이션은 플랫폼 독립적인 단위 서비스가 된다.

[그림 3]의 절차를 통해 배포된 단위 서비스들은 BPEL을 이용하여 사용자 정의 프로세스 통합이 가능하다.

[그림 4]는 기존의 애플리케이션들이 [그림 3]의 절차를 통해 웹 서비스로 배포된 후, 외부 웹 서비스 또는 모듈들과 함께 BPEL로 하나의 프로세스로 통합하기 위한 모델을 보여준다.

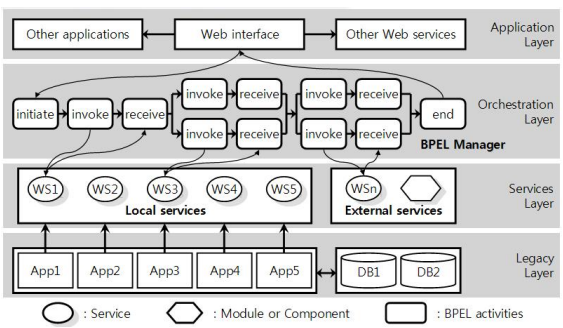

그림 4. 웹 서비스 통합 모델
Fig 4. Web service integration model

BPEL을 통해 내, 외부 단위 서비스들을 자유롭게 BPEL 프

로세스에 연결하여 사용할 수 있는데, 연결되는 서비스는 각각 invoke와 receive라는 BPEL activities와 연결된다. 각각의 서비스는 invoke activity를 통해 입력 데이터를 전달받으며, receive activity를 통해 결과 값을 다음 invoke activity로 전달함으로써 다음 서비스와 유기적으로 연결된다. 이런 과정을 거쳐 얻게 된 결과값은 표준 XML 데이터 형식이므로 별도의 가공 없이 외부의 다른 웹 서비스나 애플리케이션에서 활용 가능하다.

Ⅳ. SeqWeB의 시스템 구조 및 주해 과정

SeqWeB 시스템은 그림 5에서 보는 바와 같이 Data/Service Layer, SeqWeB Middle Layer, SeqWeB Service Access Layer로 구성된다.

4.1 Data/Services Layer (SL)

Data/Services Layer는 EST 서열 주해를 위한 분석 서비스, 데이터베이스, 외부 웹 서비스로 구성된다.

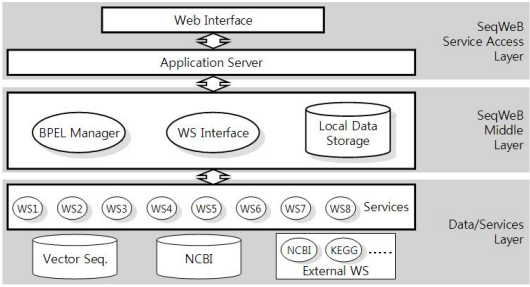

그림 5. SeqWeB 시스템 구조도
Fig 5. System architecture of SeqWeB

EST 서열 주해를 실질적으로 담당하는 분석 프로그램들은 모두 단위 웹 서비스로 제작/배포됐으며, SeqWeB에서 제공하는 분석 서비스나 분석 프로그램은 〔표 1〕 과 같다.

그림 3의 절차를 통해 웹 서비스로 배포된 표 1의 8개 분석 프로그램들은 표준 기술의 사용을 통해 플랫폼 독립성 및 통합 시 상호 운용성을 보장받을 수 있다.

분석 서비스에서 사용되는 모든 데이터베이스는 SeqWeB 서버에 로컬 데이터베이스 형태로 구축하였다. Vector Seq. 데이터베이스에는 표 1의 2단계 벡터 서열 제거 과정에서 사용되는 기존에 알려진 모든 벡터 서열과 사용자가 직접 올린 벡터 서열이 저장되어 관리된다. 또한, 6단계의 상동성 검색을 위해 NCBI 데이터베이스가 활용된다.

표 1. SeqWeB의 분석 서비스 및 분석 프로그램
Table 1. Analysis services and analytic tools of SeqWeB

단 계	분석 서비스	분석 프로그램	개발 언어
1단계	염기 호출	Phred	C
2단계	벡터 서열 제거	cross_match	C
3단계	반복 서열 제거	RepeatMasker	C
4단계	클러스터링	TGICL, ICAtools	C
5단계	어셈블링	CAP3, Phrap	C
6단계	상동성 검색	BLAST	C

3장에서 언급했듯이 웹 서비스 모델은 외부 웹 서비스와의 연동을 통해 기능적 확장이 가능하다. SeqWeB은 BLAST 결과의 Accession number를 이용하여 NCBI에서 제공하는 웹 서비스인 E-Utilities를 호출함으로써 사용자에게 더 풍부한 주해정보를 제공한다.

4.2 SeqWeB Middle Layer (ML)

SeqWeB Middle Layer는 Service Access Layer와 Data/Service Layer 중간에서 사용자와 시스템 간 상호 작용을 지원한다. Middle Layer로 본 연구에서는 Oracle BPEL Manager를 사용하였다.

SeqWeB은 시스템 구현에 JAVA를 사용했지만 분석 프로그램들은 표 1과 같이 대부분이 C나 Perl로 구현되었다. 시스템과 분석 프로그램이 각각 이질적인 프로그램 언어로 구현되었기 때문에 이 둘 간의 원활한 호출과 데이터 교환이 어렵다. 그러므로 분석 프로그램을 단위 웹 서비스로 배포하는 과정에서 시스템과의 호환을 위한 별도의 인터페이스가 필요하다. SeqWeB에서 제공하는 모든 분석 서비스는 각각의 서비스에 맞게 설계와 구현된 웹 서비스 인터페이스가 있다. 이 때문에 SL의 모든 단위 서비스들은 시스템에 의한 원활한 호출 및 데이터 교환이 가능하다.

Local Data Storage는 SeqWeB에서 이뤄지는 EST 서열 주해와 관련된 모든 데이터를 저장한다. 사용자가 주해 과정에서 입력한 모든 파라미터를 비롯하여 다수의 분석 과정에서 발생하는 처리 결과를 저장한다. 이를 통해 SeqWeB은 사용자에게 개인화된 서비스를 제공하며, 사용자는 시스템에 저장된 자신의 주해 이력을 조회할 수 있다. 또한, 저장된 모든 데이터는 XML 형식으로도 제공하므로 다른 웹 서비스나

XML을 사용하는 다른 분석 프로그램에 별도의 파싱 작업 없이 재사용 가능하다.

4.3 SeqWeB Service Access Layer (AL)

SeqWeB Service Access Layer는 사용자에게 웹 기반의 서비스 이용을 지원한다. SeqWeB의 Application Server를 통해 사용자는 인터넷 환경에서 Internet Explorer나 Firefox 등과 같은 웹 브라우저를 통해 시간적, 공간적 제약 없이 해당 서비스를 이용할 수 있으며 EST 서열 주해와 관련된 다양한 옵션을 통해 보다 양질의 주해 결과를 얻을 수 있다. 또한, 단위 서비스의 서열 데이터 처리 결과를 웹 브라우저에 제시하여 사용자는 실시간으로 처리 결과를 확인할 수 있을 뿐만 아니라 필요에 따라 결과 데이터를 수정할 수도 있다. SeqWeB의 웹 인터페이스는 JSP로 구현되었으며, Application Server는 Apache Tomcat 5.5 버전을 사용하였다.

4.4 SeqWeB의 EST 서열 주해 과정

SeqWeB의 EST 서열 주해는 (1) 염기 호출, (2) 정제, (3) 클러스터링 및 어셈블링, (4) 주해에 이르는 4개 모듈의 연속된 분석 작업이다. 그림 6은 SeqWeB의 4개 분석 모듈과 모듈 간 분석 흐름을 보여준다.

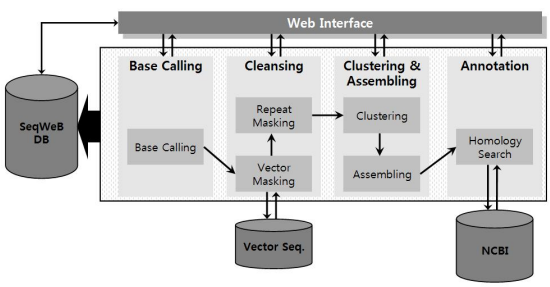

그림 6. SeqWeB의 모듈과 분석 흐름
Fig 6. Work flow of SeqWeB

4.4.1 염기 호출 (Base Calling)

염기 호출 (Base calling)은 생물학자가 얻어낸 크로마토그램을 읽어 들여 각각의 해당 염기를 결정해주는 과정으로 Phred 프로그램을 사용한다.

Phred[3]는 SCF (AVI model 373 및 377), ESD (MegaBACE) 등의 크로마토그램 파일을 자동으로 읽어서 각각의 서열 정보를 출력하며, 서열의 quality file과 quality score를 제공한다. quality file은 이후 정제와 클러스터링과 어셈블링 과정에서도 분석을 위해 계속 사용된다.

4.4.2 정제 (Cleansing)

SeqWeB에서는 EST 서열에서 불필요한 부분을 제거하기 위해 벡터 서열 제거와 반복 서열 제거 서비스를 제공한다.

SeqWeB의 벡터 서열 제거 서비스는 cross_match[19] 프로그램을 이용하여 벡터 서열을 식별하고, 염기 호출 과정에서 생성된 quality file과 최소 quality score를 통해 low quality 서열들을 제거한다. 또한, 벡터 서열의 식별을 위해 별도의 벡터 서열 데이터베이스를 이용하는데 사용자가 만약 자신만의 벡터 서열 데이터를 갖고 있다면 업로드 기능을 사용하여 벡터 서열의 식별이 가능하다.

반복 서열 제거 서비스는 RepeatMasker[20] 프로그램을 이용한다. RepeatMasker는 사용자가 선택한 반복 서열 데이터베이스에서 DNA 서열 내 단순 반복되는 염기 서열을 제거한다.

4.4.3 클러스터링과 어셈블링

EST 서열의 정제 (cleansing) 과정이 끝나면 클러스터링과 어셈블링 과정을 거친다. 이 과정의 핵심은 cDNA 라이브러리의 발현된 유전자들만을 식별하는 것이다.

SeqWeB은 클러스터링 서비스를 위하여 TGICL[4]과 ICAtools[21] 프로그램을 사용한다. 두 프로그램 모두 정제된 EST 서열의 클러스터링을 위한 똑같은 기능의 분석 프로그램으로 사용자의 선호도, 주해 종 (種)의 특성 등에 따라 선택적 사용이 가능하다. 클러스터링 과정을 거치면 기능별 서열 클러스터가 생성되고 SeqWeB은 각 클러스터의 부모 서열만을 추출한다.

클러스터링 과정이 끝나면 복수의 부모 서열을 하나 또는 복수의 컨티그 (contig) 파일로 합치는 어셈블링 과정을 거친다. 어셈블링 서비스는 CAP3[22]와 Phrap[19] 프로그램을 사용할 수 있다. 어셈블링 과정 이후 클러스터링 과정에서 추출된 부모 서열들이 하나 또는 복수의 컨티그 파일로 생성되며 이 컨티그 파일을 이용하여 상동성 검색을 수행한다.

4.4.4 주해 (Annotation)

염기 호출, 정제, 클러스터링과 어셈블링 과정을 모두 마치면 마지막으로 주해 과정이다. SeqWeB의 주해 서비스는 NCBI의 BLAST[5] 프로그램을 사용한다. SeqWeB은 BLAST의 실행을 위해 로컬 데이터베이스를 구축하여 사용하며, 현재 blastp, blastn, blastx 프로그램으로 상동성 검색이 가능하다. 사용자는 데이터베이스, e-value, 결과 포맷의 옵션을 설정할 수 있다. 그리고 NCBI에서 제공하는 E-Utilities[23] 웹 서비스와의 연동을 통해 BLAST 결과에 대해 보다 풍부한 정보의 제공이 가능하다.

V. 검증

본 장에서는 Cattle EST 서열을 이용하여 SeqWeB의 주해 성능을 검증한 후, 기존 EST 서열 주해 시스템의 통합 방식과 SeqWeB의 통합 방식 간 비교를 통해 시스템의 확장, 수정과 유지보수의 이점을 검증한다.

5.1 주해 성능 검증

본 논문에서는 University of Illinois에서 완료된 Cattle EST 프로젝트[17]의 Cattle spleen EST 서열을 이용하여 시스템 성능 검증을 수행하였다.

Cattle EST 프로젝트는 유전자 매핑과 발견 등을 위한 자료를 제공하고자 시작되었다. 특히, Placenta, Ovary, Spleen의 세 cDNA libraries가 EST 시퀀싱을 위해 만들어졌다. 현재 726개의 Cattle Spleen EST 서열이 GeneBank에 저장되어 있으며, 곧 5,000개 이상의 Spleen EST 서열이 추가로 저장될 예정이다.

[18]에서 주해가 완료된 Cattle EST 서열 파일을 다운로드할 수 있다. 본 논문에서는 Cattle Spleen cDNA에서 생산된 EST 서열 파일 638MB를 검증 데이터로 사용하였다.

표 2는 SeqWeB 서버 환경과 검증에서 적용된 주해 프로세스, 그리고 검증 데이터에 대한 정보를 보여준다.

표 2. 검증 환경
Table 2. Validation conditions

Server Performance	OS	Linux Fedora Core 7
	CPU	2.5GHz
	RAM	1GB
Analysis Procedure & Application	Base Calling	Phred
	Vector Masking	cross_match
	Repeat Masking	RepeatMasker
	Clustering	TGICL
	Assembling	CAP3
	Homology Search	BLASTN
Input Data	cDNA Library	Cattle Spleen
	Test Size	39.6MB (Random Extraction)

다운로드한 638MB의 EST 서열 가운데 임의로 39.6MB를 추출하여 SeqWeB에 업로드한 후, SeqWeB에서 제공하는 염기 호출 (Phred) - 벡터 서열 제거 (cross_match) - 반복 서열 제거 (RepeatMasker) - 클러스터링 (TGICL) - 어셈블링 (CAP3) - 상동성 검색 (BLAST) 서비스를 이용하여 입력한 EST 서열의 주해를 수행하였다. 주해 결과는 [표 3]과 같다.

표 3. Cattle spleen EST 서열 주해 결과 요약
Table 3. Validation summary

구분	내 용	크기
입력 데이터	164개 크로마토그램	39.6MB
염기 호출 (Phred)	164개의 서열 파일 생성 전체 길이: 128,460bp	13.8MB
벡터서열제거 (cross_match)	서열 개수: 164개 처리 후: 123,208bp (5,252bp masked)	13.7MB
반복서열제거 (RepeatMasker)	서열 개수: 164개 처리 후: 102,034bp (10,587bp masked)	13.6MB
클러스터링 (TGICL)	서열 개수: 2개	1.64KB
어셈블링 (CAP3)	컨티그 개수: 1개	0.8MB
상동성 검색 (BLAST)	프로그램: blastn 데이터베이스: est_others e-value threshold: 1e-4	1MB
결 과	Lewin Cattle Spleen Bos taurus cDNA clone	-

입력된 128,460bp의 서열이 벡터 및 반복 서열 제거 과정을 거치면서 총 15,839bp가 제거되어 102,034bp의 정제된 서열로 처리되었다. 이 서열에 대하여 클러스터링 분석을 실행한 결과 총 2개의 서열이 기능별 클러스터의 부모 서열로 판명되었으며, 2개의 서열에 대한 어셈블링 과정을 거쳐 1개의 컨티그 서열을 확보하였다. 컨티그 서열을 blastn 프로그램을 이용하여 est_others 데이터베이스에 상동성 검색을 수행한 결과 13분 후에 웹 브라우저를 통한 한번의 데이터 입력만으로 입력 서열에 대한 정확한 주해 결과를 확인할 수 있었다.

5.2 SeqWeB 시스템의 구조적 이점 검증

다수의 애플리케이션 통합을 위해 SOA를 도입함으로써 구축에 따르는 비용, 시간과 사후 유지보수 관리 비용에 있어서 커다란 효과를 기대할 수 있다. 본 절에서는 Point-to-Point 통합 방식과 SOA 기반의 통합 방식을 비교함으로써 본 논문에서 제안하는 시스템의 확장, 수정 및 유지보수 관리의 이점을 검증한다.

표 4는 어플리케이션들을 통합하는 방식에 있어 Point-to-Point 통합 방식과 SOA 기반의 통합 방식을 비교한 것이다.

표 4. 통합 방식 간 비교
Table 4. Comparison of two integration methods

구분	Point-to-Point	SOA
통합 레벨	어플리케이션 통합	프로세스 통합
기술	벤더 종속적 기술	표준 기술
구조	분산된 어플리케이션들의 강 결합	버스 형태의 약 결합 구조
통합범위	최소 단위 어플리케이션 간 통합	단위 시스템 간 전사적 통합
경제성 (TCO)	사후 지속적 추가 비용 발생	사후 추가 비용 거의 들지 않음

Point-to-Point 통합 방식은 가장 기초적인 애플리케이션 통합 방법으로써 벤더 종속적 기술로 개발된 인터페이스를 통해 1:1 방식으로 통합한다. 통합된 애플리케이션에 대하여 추가 확장이나 변경, 유지보수를 수행하기 위해서는 추가 인터페이스의 개발 또는 기존 인터페이스의 수정을 위한 추가의 비용, 시간, 자원이 소요된다.

SOA 기반의 통합 방식은 현재 가장 발전된 형태의 IT 인프라 통합 방법으로써 표준 XML 등의 기술을 이용하여 프로세스 수준의 통합을 수행한다. 또한 버스를 통한 약 결합을 수행함으로써 추가 확장이나 변경, 유지보수에 추가 비용이 거의 들지 않는다.

표 5는 표 4의 두 통합 방식에 따른 인터페이스 개발 건수를 비교한 것이다.

표 5. 통합 방식에 따른 인터페이스 개발 건수
Table 5. The number of interfaces according to two integration methods

개발 항목	적용항목	Point-to-Point	SOA
일반	-	5	2
염기 호출	Phred	2	1
벡터서열제거	cross_match	2	1
반복서열제거	RepeatMasker	2	1
클러스터링	TGICL	2	1
	ICAtools	2	1
어셈블링	CAP3	2	1
	Phrap	2	1
상동성 검색	BLAST	2	1
계		21	10

인터페이스 개발 건수 산출에서 웹 서비스 배포 시 툴에 의해 자동으로 생성되는 WSDL 문서와 두 통합 유형에 모두 동일하게 개발되어야 하는 인터페이스는 제외되었다. 각 통합 방식별 필요한 인터페이스의 총 개수를 보면 SOA 기반의 통합 방식이 Point-to-Point 통합 방식에 비해 52.4% 적은 인터페이스의 개발 건수를 보여준다.

그림 7은 각 통합 방식별로 10개의 분석 프로그램을 추가함으로써 새롭게 개발되어야하는 인터페이스 개수의 증가량을 보여준다.

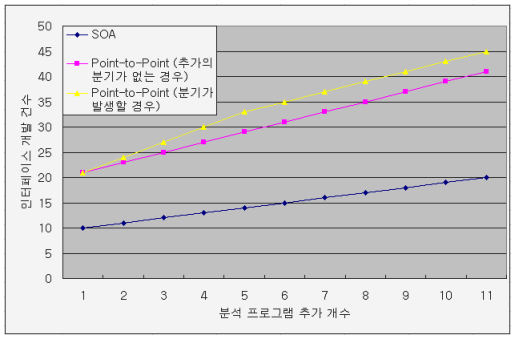

그림 7. 확장 시 인터페이스 개발 건수
Fig 7. The number of newly developed interfaces according to system extension

그림 7에서 보는바와 같이 SOA 기반 통합 방식의 경우 10개의 분석 프로그램을 추가할 경우 표 5와 같이 각 1개의 인터페이스 개발로 총 10개의 인터페이스 개수가 증가한다. 반면 Point-to-Point 통합 방식의 경우 추가의 분기가 발생하지 않는 경우, 즉 클러스터링과 어셈블링의 분석 과정에만 분석 프로그램이 추가될 경우 가운데 선과 같은 증가량을 보이며, 추가의 분기가 발생할 경우 더 많은 인터페이스가 개발되어야함을 보여준다.

그림 7이 분석 프로그램의 추가에 따른 새로운 인터페이스의 개발 건수를 비교했다면, 그림 8은 분석 프로그램이 추가될 경우 수정되어야하는 기존 인터페이스 개수의 증가량을 보여준다.

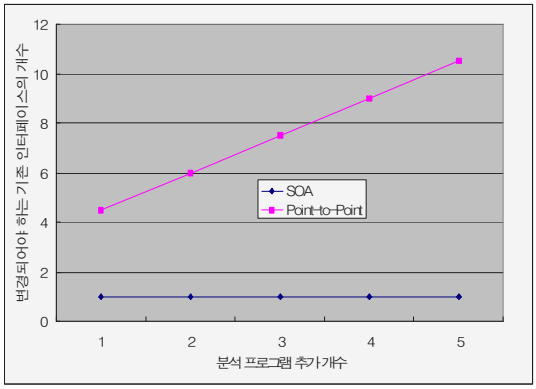

그림 8. 확장 시 기존 인터페이스의 수정 건수
Fig 8. The number of modified interfaces according to system extension

SOA 기반 통합 방식의 경우 분석 프로그램의 추가 개수에 관계없이 조합을 관리하는 BPEL 인터페이스 1개만 수정이 필요한 반면, Point-to-Point 통합 방식의 경우 비교적 많은 기존의 인터페이스가 수정되어야 함을 알 수 있다.

인터페이스의 개발 건수 또는 수정 건수가 많다는 것은 비용, 시간, 인력 등의 자원이 더 많이 소요됨을 의미하고, 이는 곧 사후 유지보수 비용의 증가로 직결된다. 검증 결과를 통해 본 논문에서는 SOA 기반의 통합 방식이 기존의 통합 방식에 비해 확장 및 수정을 비롯하여 유지보수가 더욱 용이함을 증명하였다.

V. 결론 및 향후 연구

본 논문에서는 SOA 기반의 웹 서비스와 BPEL 기술을 이용한 EST 서열 주해 시스템 SeqWeB을 제안한다. SeqWeB은 SOA 기반의 8개 분석 서비스를 통합한 서비스 통합 계층의 시스템으로 확장성이 뛰어나고 유연한 구조를 갖고 있다. XML, WSDL, SOAP 등과 같은 표준 기술을 사용하여 서비스 간 상호 운용성을 보장할 뿐만 아니라 선택적 사용이 가능하다. 또한, 개인화된 서비스를 제공하며, 웹 브라우저를 통해 손쉽게 시스템과 상호작용을 할 수 있다. SeqWeB에 적용된 웹 서비스 제작 기법 및 다수의 서비스 통합 방법론은 생물학 관련 분석 서비스들의 효율적인 통합을 위한 새로운 방법론을 제시하였다.

SeqWeB은 EST 서열 주해 결과를 이용하여 진행 가능한 Genome 분석이나 대사경로 분석 등과 같은 생물학 프로세스와 통합이 가능한 인프라 및 통합 방법론을 마련했다는데 의의가 있다.

참고문헌

[1] Adams,M..D., et al(1991). Complementary DNA sequencing: expressed sequence tags and human genome project. Science, 252, pp.1651-1661.

[2] Parkinson,J., et al(2004). Parasite Genome Protocols Humana Press, Totowa,N.J.

[3] Ewing,B., et al(1998). Base-calling of automated sequencer traces using phred. I. Accuracy assessment. Genome Res, 8, pp.175 - 185.

[4] Pertea,G., et al(2003). TIGR Gene Indices Clustering Tools (TGICL): a software system for fast clustering of large EST datasets. Bioinformatics, 19, 651 - .652.

[5] Altschul,S.F., et al(1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res., 25, 3389 - .3402.

[6] Kunne,C., et al(2005). CR-EST: a resource for crop ESTs. Nucleic Acids Res., 33, D619 - .D621.

[7] Mao,C., et al(2003). ESTAP - .an automated system for the analysis of EST data. Bioinformatics, 19, 1720 - 1722.

〔8〕 Hotz-Wagenblatt,A., et al(2003). ESTAnnotator : a tool for high throughput EST annotation. Nucleic Acids Res., 31, 3716 - .3719.
〔9〕 http://www.ensoltek.co.kr/Products/Products _geneF .htm
〔10〕 Nagaraj,S.H., et al(2007). A hitchhiker's guide to expressed sequence tag (EST) analysis. Brief. Bioinform., 8, 6-21.
〔11〕 Byungwook Lee., et al(2007). ESTpass: a web -based server for processing and annotating expressed sequence tag (EST) sequences. Nucleic Acids Research, Vol 35. W159-W162.
〔12〕 이경하, et al(2004). SOA (Service-Oriented Architecture)와 웹 서비스. 한국정보과학회, 정보과학회지 제22권 제10호. pages 5-10.
〔13〕 http://www.w3.org/
〔14〕 S. Carrere, et al(2006). Remora: a pilot in the ocean of biomoby web-services. Bioinformatics, 22(7).
〔15〕 Xiaorong Xiang, et al(2007). A Service-oriented Data Integration and Analysis Environment for In Silico Experiments and Bioinformatics Research. Proceedings of the 40thAnnual Hawaii International Conference on System Sciences.
〔16〕 http://en.wikipedia.org/wiki/BPEL
〔17〕 Larson JH, et al(2006). Genomic organization and evolution of the ULBP genes in cattle. BMC Genomics. 7:227.
〔18〕 http://titan.biotec.uiuc.edu/cattle/cattle_ project .htm
〔19〕 http://www.phrap.org/phredphrapconsed.html
〔20〕 http://www.repeatmasker.org/
〔21〕 http://www.littlest.co.uk/software/bioinf/old_packages /icatools/
〔22〕 Huang,X. and Madan,A. (1999) CAP3: A DNA sequence assembly program. Genome Res., 9, 868 - .877.
〔23〕 http://www.ncbi.nlm.nih.gov/

저 자 소 개

남 성 혁

2006년 2월 : 충북대학교 경영정보학과 학사
2008년 2월 : 충북대학교 바이오정보기술학과 석사
2008년 3월 ~ 현재 : 한국생명공학연구원/UST 기능유전체학 박사 과정
관심분야 : 바이오 인포메틱스, SOA, BPM, Web services, XML

김 태 경

2002년 2월 : 충북대학교 경영정보학과 학사
2005년 2월 : 충북대학교 정보산업공학과 석사
2005년 2월 ~ 현재 : 충북대학교정보산업공학과 박사 과정
관심분야 : 바이오인포메틱스, 그리드 컴퓨팅, XML, 데이터웨어하우스

김 경 란

2004년 2월 : 충북대학교 경영정보학과 학사
2006년 2월 : 충북대학교 정보산업공학과 석사
2006년 9월 ~ 현재 : 충북대학교 경영정보학과 박사 과정
관심분야 : 바이오인 포매틱스, 데이터웨어하우스, ERP, BI

조 완 섭

1997년 2월 ~ 현재 : 충북대학교 경영정보과 교수
1987년 2월 ~ 1990년 12월 : 전자통신연구원 연구원
2001년 ~ 2002년 : University of Florida, Post-Doc.
1996년 2월 : 한국과학기술원 전산학과 박사
관심분야 : SOA, BPM, DW & OLAP, 데이터 마이닝, 바이오 정보시스템, CRM, 전자상거래