(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0105325
(43) 공개일자 2017년09월19일

(51) 국제특허분류(Int. Cl.)
G06F 9/44 (2006.01) *G06F 15/16* (2006.01)
G06F 17/24 (2006.01) *G06F 17/28* (2006.01)
(52) CPC특허분류
G06F 8/10 (2013.01)
G06F 15/16 (2013.01)
(21) 출원번호 10-2016-0028409
(22) 출원일자 2016년03월09일
심사청구일자 2016년03월09일

(71) 출원인
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
노재춘
서울특별시 도봉구 방학로15길 40, 107동 701호 (방학동, 신동아아파트)
김현지
경기도 성남시 분당구 구미로 115, 604동 802호(구미동, 무지개마을건영아파트)
김영우
경기도 성남시 분당구 정자로 112, 504동 903호 (정자동, 정든마을신화5단지아파트)
(74) 대리인
두호특허법인

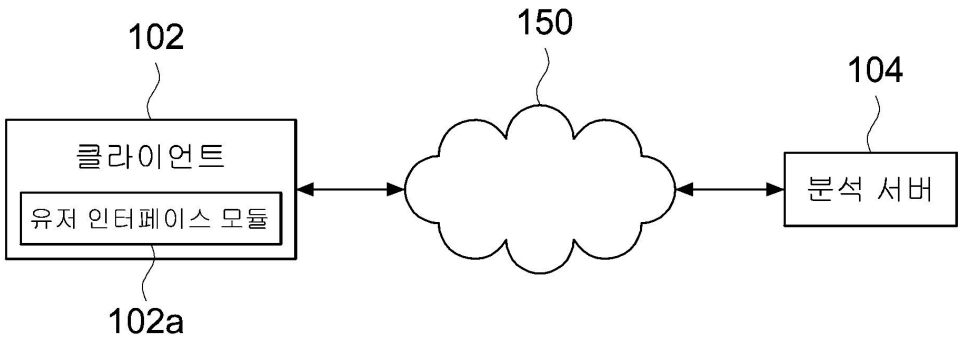
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 **소스 코드 분석 방법과 이를 수행하기 위한 시스템 및 장치**

(57) 요 약

소스 코드 분석 방법과 이를 수행하기 위한 시스템 및 장치가 개시된다. 예시적인 실시 예에 따른 소스 코드 분석 시스템은, 웹 유저 인터페이스를 통해 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 포함하는 소스 코드 분석 요청을 전송하는 클라이언트 및 소스 코드 분석 요청을 수신하고, 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하며, 추출한 기본 설정 파일을 설정 정보에 따라 변경하여 실행 설정 파일을 생성하고, 생성한 실행 설정 파일로 소스 코드 파일을 분석하기 위한 적어도 하나의 소프트웨어를 실행시켜 분석 결과 파일을 생성하며, 생성한 분석 결과 파일을 상기 클라이언트로 전달하는 분석 서버를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 17/241 (2013.01)
G06F 17/289 (2013.01)
G06F 8/75 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711013408
부처명 미래창조과학부
연구관리전문기관 한국연구재단
연구사업명 중견연구자지원사업
연구과제명 데스크탑 가상화의 성능 향상을 위한 계층형 이기종 멀티 SSD 파티션 기반의 선택적 VM 데이터 매핑과 자원 재분배에 관한 연구(NRF-2014R1A2A2A01002614)
기 여 율 1/1
주관기관 세종대학교 산학협력단
연구기간 2014.05.01 ~ 2017.04.30

명 세 서

청구범위

청구항 1

웹 유저 인터페이스를 통해 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 포함하는 소스 코드 분석 요청을 전송하는 클라이언트; 및

상기 소스 코드 분석 요청을 수신하고, 상기 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하며, 상기 추출한 기본 설정 파일을 상기 설정 정보에 따라 변경하여 실행 설정 파일을 생성하고, 상기 생성한 실행 설정 파일로 상기 소스 코드 파일을 분석하기 위한 적어도 하나의 소프트웨어를 실행시켜 분석 결과 파일을 생성하며, 생성한 상기 분석 결과 파일을 상기 클라이언트로 전달하는 분석 서버를 포함하는, 소스 코드 분석 시스템.

청구항 2

청구항 1에 있어서,

상기 분석 서버는,

클라우드 기반의 서버로서, 프로그래밍 언어 별로 기본 설정 파일을 각각 저장하는, 소스 코드 분석 시스템.

청구항 3

청구항 2에 있어서,

상기 분석 서버는,

상기 수신한 소스 코드 분석 요청에 포함된 상기 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하며, 상기 추출한 기본 설정 파일을 상기 설정 정보에 따라 변경하여 실행 설정 파일을 생성하는 설정 파일 생성 모듈;

상기 생성한 실행 설정 파일로 적어도 하나의 연동된 오픈 소스 소프트웨어를 실행시켜 상기 분석 결과 파일을 생성하는 소스 분석 모듈; 및

상기 분석 결과 파일을 상기 클라이언트로 전송하는 분석 관리 모듈을 포함하는, 소스 코드 분석 시스템.

청구항 4

청구항 3에 있어서,

상기 소스 분석 모듈은,

상기 소스 코드 파일에 포함되는 소스 코드들에 대해 해당 소스 코드를 설명하는 주석 처리를 하여 소스 코드 주석 문서를 생성하는 제1 오픈 소스 소프트웨어 및 상기 제1 오픈 소스 소프트웨어와 연동하여 동작하고 상기 소스 코드 파일에 포함되는 컴포넌트들 간의 경로, 상기 컴포넌트를 설정하거나 읽을 수 있는 경로, 및 상기 컴포넌트들 간의 관계도 중 적어도 하나를 시각화하여 표시하는 제2 오픈 소스 소프트웨어 중 적어도 하나를 실행시키는, 소스 코드 분석 시스템.

청구항 5

청구항 4에 있어서,

상기 소스 분석 모듈은,

상기 제1 오픈 소스 소프트웨어가 자국어를 지원하지 않는 경우, 상기 소스 코드 주석 문서를 유니코드로 변환하고, 상기 유니코드를 자국어로 텍스트 변환하는 제3 오픈 소스 소프트웨어를 실행시키는, 소스 코드 분석 시스템.

청구항 6

하나 이상의 프로세서들, 및

상기 하나 이상의 프로세서들에 의해 실행되는 하나 이상의 프로그램들을 저장하는 메모리를 구비한 컴퓨팅 장치에서 수행되는 방법으로서,

클라이언트로부터 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 포함하는 소스 코드 분석 요청을 수신하는 단계;

상기 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하는 단계;

상기 추출한 기본 설정 파일을 상기 설정 정보에 따라 변경하여 실행 설정 파일을 생성하는 단계;

상기 생성한 실행 설정 파일로 상기 소스 코드 파일을 분석하기 위한 적어도 하나의 소프트웨어를 실행시켜 분석 결과 파일을 생성하는 단계; 및

상기 생성한 분석 결과 파일을 상기 클라이언트로 전송하는 단계를 포함하는, 소스 코드 분석 방법.

청구항 7

청구항 6에 있어서,

상기 컴퓨팅 장치는,

클라우드 기반의 서버로서, 프로그래밍 언어 별로 기본 설정 파일을 각각 저장하는, 소스 코드 분석 방법.

청구항 8

청구항 6에 있어서,

상기 분석 결과 파일을 생성하는 단계는,

상기 소스 코드 파일에 포함되는 소스 코드들에 대해 해당 소스 코드를 설명하는 주석 처리를 하여 소스 코드 주석 문서를 생성하는 제1 오픈 소스 소프트웨어를 실행시키는 단계; 및

상기 제1 오픈 소스 소프트웨어와 연동하여 동작하고 상기 소스 코드 파일에 포함되는 컴포넌트들 간의 경로, 상기 컴포넌트를 설정하거나 읽을 수 있는 경로, 및 상기 컴포넌트들 간의 관계도 중 적어도 하나를 시각화하여 표시하는 제2 오픈 소스 소프트웨어를 실행시키는 단계를 포함하는, 소스 코드 분석 방법.

청구항 9

청구항 8에 있어서,

상기 분석 결과 파일을 생성하는 단계는,

상기 제1 오픈 소스 소프트웨어가 자국어를 지원하는지 여부를 확인하는 단계; 및

상기 자국어를 지원하지 않는 경우, 상기 소스 코드 주석 문서를 유니코드로 변환하고, 상기 유니코드를 자국어로 텍스트 변환하는 제3 오픈 소스 소프트웨어를 실행시키는 단계를 더 포함하는, 소스 코드 분석 방법.

청구항 10

하나 이상의 프로세서들;

메모리; 및

하나 이상의 프로그램들을 포함하고,

상기 하나 이상의 프로그램들은 상기 메모리에 저장되고, 상기 하나 이상의 프로세서들에 의해 실행되도록 구성되며,

상기 하나 이상의 프로그램들은,

클라이언트로부터 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 포함하는 소스 코드 분석 요청을 수신하기 위한 명령;

상기 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하기 위한 명령;

상기 추출한 기본 설정 파일을 상기 설정 정보에 따라 변경하여 실행 설정 파일을 생성하기 위한 명령;

상기 생성한 실행 설정 파일로 상기 소스 코드 파일을 분석하기 위한 적어도 하나의 소프트웨어를 실행시켜 분석 결과 파일을 생성하기 위한 명령; 및

상기 생성한 분석 결과 파일을 상기 클라이언트로 전송하기 위한 명령을 포함하는, 컴퓨팅 장치.

청구항 11

청구항 10에 있어서,

상기 컴퓨팅 장치는,

클라우드 기반의 서버로서, 프로그래밍 언어 별로 기본 설정 파일을 각각 저장하는, 컴퓨팅 장치.

청구항 12

청구항 10에 있어서,

상기 분석 결과 파일을 생성하기 위한 명령은,

상기 소스 코드 파일에 포함되는 소스 코드들에 대해 해당 소스 코드를 설명하는 주석 처리를 하여 소스 코드 주석 문서를 생성하는 제1 오픈 소스 소프트웨어를 실행시키기 위한 명령; 및

상기 제1 오픈 소스 소프트웨어와 연동하여 동작하고 상기 소스 코드 파일에 포함되는 컴포넌트들 간의 경로, 상기 컴포넌트를 설정하거나 읽을 수 있는 경로, 및 상기 컴포넌트들 간의 관계도 중 적어도 하나를 시각화하여 표시하는 제2 오픈 소스 소프트웨어를 실행시키기 위한 명령을 포함하는, 컴퓨팅 장치.

청구항 13

청구항 12에 있어서,

상기 분석 결과 파일을 생성하기 위한 명령은,

상기 제1 오픈 소스 소프트웨어가 자국어를 지원하는지 여부를 확인하기 위한 명령; 및

상기 자국어를 지원하지 않는 경우, 상기 소스 코드 주석 문서를 유니코드로 변환하고, 상기 유니코드를 자국어로 텍스트 변환하는 제3 오픈 소스 소프트웨어를 실행시키기 위한 명령을 더 포함하는, 컴퓨팅 장치.

발명의 설명

기 술 분 야

[0001] 본 발명의 실시 예는 소스 분석 기술과 관련된다.

배 경 기 술

[0003] 최근, 오픈 소스(Open Source)(무상으로 공개되는 소스 코드 또는 소프트웨어를 의미함)의 중요성이 대두됨에 따라 많은 오픈 소스들이 시장에 쏟아지고 있으며, 이를 활용하기 위해서는 소스 분석이 필연적이다. 즉, 오픈 소스 소프트웨어를 도입하는 경우, 소프트웨어 개발자는 오픈 소스의 소스 코드를 분석하여 소스 코드 구조를 파악해야 하고, 소스 코드 내에 사용된 자료 구조나 함수의 역할 등을 숙지하여야 한다.

[0004] 그러나, 기존에는 대부분 CLI(Command Line Interface) 환경에서 코드를 분석하기 때문에 코드 분석에 상당히 많은 노력과 시간이 요구된다. 또한, 기존의 소스 코드를 분석하고 가시화하는 프로그램(예를 들어, Magnum, Visustin, Soureinsight 등)들은 지원하는 프로그래밍 언어가 제한되어 있고, 소스 코드 규모가 커지면서 시스템 부하가 가중되거나 다운되는 등의 문제가 있어, 리눅스(Linux), 오픈스택(OpenStack)과 같은 대형 프로젝트 개발자가 대용량의 코드를 분석하는 경우에 적합하지 않게 된다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 한국등록특허공보 제10-0501936호(2005.07.18)

발명의 내용

해결하려는 과제

[0007] 본 발명의 실시 예는 대용량의 코드를 신속하게 분석할 수 있는 소스 코드 분석 방법과 이를 수행하기 위한 시스템 및 장치를 제공하기 위한 것이다.

과제의 해결 수단

[0009] 예시적인 실시 예에 따른 소스 코드 분석 시스템은, 웹 유저 인터페이스를 통해 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 포함하는 소스 코드 분석 요청을 전송하는 클라이언트; 및 상기 소스 코드 분석 요청을 수신하고, 상기 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하며, 상기 추출한 기본 설정 파일을 상기 설정 정보에 따라 변경하여 실행 설정 파일을 생성하고, 상기 생성한 실행 설정 파일로 상기 소스 코드 파일을 분석하기 위한 적어도 하나의 소프트웨어를 실행시켜 분석 결과 파일을 생성하며, 생성한 상기 분석 결과 파일을 상기 클라이언트로 전달하는 분석 서버를 포함한다.

[0010] 상기 분석 서버는, 클라우드 기반의 서버로서, 프로그래밍 언어 별로 기본 설정 파일을 각각 저장할 수 있다.

[0011] 상기 분석 서버는, 상기 수신한 소스 코드 분석 요청에 포함된 상기 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하며, 상기 추출한 기본 설정 파일을 상기 설정 정보에 따라 변경하여 실행 설정 파일을 생성하는 설정 파일 생성 모듈; 상기 생성한 실행 설정 파일로 적어도 하나의 연동된 오픈 소스 소프트웨어를 실행시켜 상기 분석 결과 파일을 생성하는 소스 분석 모듈; 및 상기 분석 결과 파일을 상기 클라이언트로 전송하는 분석 관리 모듈을 포함할 수 있다.

[0012] 상기 소스 분석 모듈은, 상기 소스 코드 파일에 포함되는 소스 코드들에 대해 해당 소스 코드를 설명하는 주석 처리를 하여 소스 코드 주석 문서를 생성하는 제1 오픈 소스 소프트웨어 및 상기 제1 오픈 소스 소프트웨어와 연동하여 동작하고 상기 소스 코드 파일에 포함되는 컴포넌트들 간의 경로, 상기 컴포넌트를 설정하거나 읽을 수 있는 경로, 및 상기 컴포넌트들 간의 관계도 중 적어도 하나를 시각화하여 표시하는 제2 오픈 소스 소프트웨어 중 적어도 하나를 실행시킬 수 있다.

[0013] 상기 소스 분석 모듈은, 상기 제1 오픈 소스 소프트웨어가 자국어를 지원하지 않는 경우, 상기 소스 코드 주석 문서를 유니코드로 변환하고, 상기 유니코드를 자국어로 텍스트 변환하는 제3 오픈 소스 소프트웨어를 실행시킬 수 있다.

[0014] 예시적인 실시 예에 따른 소스 코드 분석 방법은, 하나 이상의 프로세서들, 및 상기 하나 이상의 프로세서들에 의해 실행되는 하나 이상의 프로그램들을 저장하는 메모리를 구비한 컴퓨팅 장치에서 수행되는 방법으로서, 클라이언트로부터 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 포함하는 소스 코드 분석 요청을 수신하는 단계; 상기 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하는 단계; 상기 추출한 기본 설정 파일을 상기 설정 정보에 따라 변경하여 실행 설정 파일을 생성하는 단계; 상기 생성한 실행 설정 파일로 상기 소스 코드 파일을 분석하기 위한 적어도 하나의 소프트웨어를 실행시켜 분석 결과 파일을 생성하는 단계; 및 상기 생성한 분석 결과 파일을 상기 클라이언트로 전송하는 단계를 포함한다.

[0015] 상기 컴퓨팅 장치는, 클라우드 기반의 서버로서, 프로그래밍 언어 별로 기본 설정 파일을 각각 저장할 수 있다.

[0016] 상기 분석 결과 파일을 생성하는 단계는, 상기 소스 코드 파일에 포함되는 소스 코드들에 대해 해당 소스 코드를 설명하는 주석 처리를 하여 소스 코드 주석 문서를 생성하는 제1 오픈 소스 소프트웨어를 실행시키는 단계; 및 상기 제1 오픈 소스 소프트웨어와 연동하여 동작하고 상기 소스 코드 파일에 포함되는 컴포넌트들 간의 경로, 상기 컴포넌트를 설정하거나 읽을 수 있는 경로, 및 상기 컴포넌트들 간의 관계도 중 적어도 하나를 시각화하여 표시하는 제2 오픈 소스 소프트웨어를 실행시키는 단계를 포함할 수 있다.

[0017] 상기 분석 결과 파일을 생성하는 단계는, 상기 제1 오픈 소스 소프트웨어가 자국어를 지원하는지 여부를 확인하는 단계; 및 상기 자국어를 지원하지 않는 경우, 상기 소스 코드 주석 문서를 유니코드로 변환하고, 상기 유니코드를 자국어로 텍스트 변환하는 제3 오픈 소스 소프트웨어를 실행시키는 단계를 더 포함할 수 있다.

[0018] 예시적인 실시 예에 따른 컴퓨팅 장치는, 하나 이상의 프로세서들; 메모리; 및 하나 이상의 프로그램들을 포함하고, 상기 하나 이상의 프로그램들은 상기 메모리에 저장되고, 상기 하나 이상의 프로세서들에 의해 실행되도록 구성되며, 상기 하나 이상의 프로그램들은, 클라이언트로부터 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 포함하는 소스 코드 분석 요청을 수신하기 위한 명령; 상기 프로그래밍 언어 정보에 기반하여 기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출하기 위한 명령; 상기 추출한 기본 설정 파일을 상기 설정 정보에 따라 변경하여 실행 설정 파일을 생성하기 위한 명령; 상기 생성한 실행 설정 파일로 상기 소스 코드 파일을 분석하기 위한 적어도 하나의 소프트웨어를 실행시켜 분석 결과 파일을 생성하기 위한 명령; 및 상기 생성한 분석 결과 파일을 상기 클라이언트로 전송하기 위한 명령을 포함한다.

[0019] 상기 컴퓨팅 장치는, 클라우드 기반의 서버로서, 프로그래밍 언어 별로 기본 설정 파일을 각각 저장할 수 있다.

[0020] 상기 분석 결과 파일을 생성하기 위한 명령은, 상기 소스 코드 파일에 포함되는 소스 코드들에 대해 해당 소스 코드를 설명하는 주석 처리를 하여 소스 코드 주석 문서를 생성하는 제1 오픈 소스 소프트웨어를 실행시키기 위한 명령; 및 상기 제1 오픈 소스 소프트웨어와 연동하여 동작하고 상기 소스 코드 파일에 포함되는 컴포넌트들 간의 경로, 상기 컴포넌트를 설정하거나 읽을 수 있는 경로, 및 상기 컴포넌트들 간의 관계도 중 적어도 하나를 시각화하여 표시하는 제2 오픈 소스 소프트웨어를 실행시키기 위한 명령을 포함한다.

[0021] 상기 분석 결과 파일을 생성하기 위한 명령은, 상기 제1 오픈 소스 소프트웨어가 자국어를 지원하는지 여부를 확인하기 위한 명령; 및 상기 자국어를 지원하지 않는 경우, 상기 소스 코드 주석 문서를 유니코드로 변환하고, 상기 유니코드를 자국어로 텍스트 변환하는 제3 오픈 소스 소프트웨어를 실행시키기 위한 명령을 더 포함할 수 있다.

발명의 효과

[0023] 본 발명의 실시 예에 의하면, 클라우드 기반의 소스 코드 분석 서비스를 제공함으로써, 코드 분석 실행 시 분석 수행 시간을 줄일 수 있으며, 그로 인해 대용량의 코드 분석을 신속하게 수행할 수 있게 된다. 또한, 프로그래밍 언어 별로 코드 분석 시 필요한 실행 설정 파일을 자동으로 생성하여 사용자의 편의성을 향상시킬 수 있게 된다.

도면의 간단한 설명

[0025] 도 1은 본 발명의 예시적인 실시 예에 따른 소스 코드 분석 시스템의 구성을 나타낸 도면

도 2는 본 발명의 예시적인 실시 예에 따른 분석 서버의 구성을 나타낸 블록도

도 3은 본 발명의 예시적인 실시 예에 따른 소스 코드 분석 방법을 나타낸 흐름도

도 4는 예시적인 실시 예에 따른 분석 서버에서 소스 코드 파일의 분석 결과를 웹 페이지에 표시하는 상태를 나타낸 도면

도 5는 예시적인 실시 예들에서 사용되기에 적합한 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도

발명을 실시하기 위한 구체적인 내용

[0026] 이하, 도면을 참조하여 본 발명의 구체적인 실시형태를 설명하기로 한다. 이하의 상세한 설명은 본 명세서에서 기술된 방법, 장치 및/또는 시스템에 대한 포괄적인 이해를 돕기 위해 제공된다. 그러나 이는 예시에 불과하며 본 발명은 이에 제한되지 않는다.

[0027] 본 발명의 실시 예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다. 상세한 설명에서 사용되는 용어는 단지 본 발명의 실시 예들을 기술하기 위한 것이며, 결코 제한적이어서는 안 된다. 명확하게 달리 사용되지 않는 한, 단수 형태의 표현은 복수 형태의 의미를 포함한다. 본 설명에서, "포함" 또는 "구비"와 같은 표현은 어떤 특성들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 특성, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다.

[0028] 이하의 설명에 있어서, 신호 또는 정보의 "전송", "통신", "송신", "수신" 기타 이와 유사한 의미의 용어는 일 구성요소에서 다른 구성요소로 신호 또는 정보가 직접 전달되는 것뿐만이 아니라 다른 구성요소를 거쳐 전달되는 것도 포함한다. 특히 신호 또는 정보를 일 구성요소로 "전송" 또는 "송신"한다는 것은 그 신호 또는 정보의 최종 목적지를 지시하는 것이고 직접적인 목적지를 의미하는 것이 아니다. 이는 신호 또는 정보의 "수신"에 있어서도 동일하다. 또한 본 명세서에 있어서, 2 이상의 데이터 또는 정보가 "관련"된다는 것은 하나의 데이터(또는 정보)를 획득하면, 그에 기초하여 다른 데이터(또는 정보)의 적어도 일부를 획득할 수 있음을 의미한다.

[0030] 도 1은 본 발명의 예시적인 실시 예에 따른 소스 코드 분석 시스템의 구성을 나타낸 도면이다.

[0031] 도 1을 참조하면, 소스 코드 분석 시스템(100)은 클라이언트(102) 및 분석 서버(104)를 포함할 수 있다. 여기서, 클라이언트(102)는 로컬 영역 네트워크(Local Area Network: LAN), 광역 네트워크(Wide Area Network: WAN), 셀룰라 네트워크 또는 인터넷과 같은 네트워크(150)를 통해 분석 서버(104)와 통신 가능하게 연결된다.

[0032] 클라이언트(102)는 네트워크(150)를 통해 분석 서버(104)와 접속하기 위한 유저 인터페이스 모듈(102a)을 포함할 수 있다. 예시적인 실시예에서, 유저 인터페이스 모듈(102a)은 웹 유저 인터페이스(Web User Interface)일 수 있다. 클라이언트(102)의 예들은 휴대폰, 스마트 폰, 태블릿, 랩톱 컴퓨터, 데스크탑 컴퓨터, 웨어러블 기기 등이 될 수 있으나, 이에 한정되는 것은 아니며 분석 서버(104)에 접속 가능한 다양한 유형의 무선 통신 장치 및 유선 통신 장치를 포함한다.

[0033] 클라이언트(102)는 사용자 식별 정보(예를 들어, 사용자 아이디, 패스워드, 이메일 주소 등)를 분석 서버(104)로 전송하여 사용자 등록을 수행할 수 있다. 클라이언트(102)는 분석하고자 하는 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 포함하는 소스 코드 분석 요청을 분석 서버(104)로 전송할 수 있다. 클라이언트(102)는 분석 서버(104)에 접속하여 상기 소스 코드 파일에 대한 분석 결과 파일을 제공받을 수 있다. 도 1에서는 클라이언트(102)가 하나인 것으로 도시하였으나, 이에 한정되는 것은 아니며 소스 코드를 분석하고자 하는 각 사용자의 클라이언트(102)가 분석 서버(104)에 접속할 수 있다.

[0034] 분석 서버(104)는 적어도 하나의 웹 페이지를 포함하는 웹 사이트를 호스팅할 수 있고, 무엇보다도 클라이언트(102)가 소스 코드 파일의 분석을 요청하고 분석 결과를 제공받기 위해 액세스할 수 있는 웹 페이지를 네트워크(150)를 통해 제공할 수 있다. 예시적인 실시 예에서, 분석 서버(104)는 클라우드 기반의 서버일 수 있다. 즉, 분석 서버(104)는 클라우드 기반의 소스 코드 분석 서비스를 제공할 수 있다.

[0035] 분석 서버(104)는 각 클라이언트(102)로부터 수신한 소스 코드 파일을 분석하여 분석 결과 파일을 생성하며, 생성한 분석 결과 파일을 해당 사용자의 디렉토리(Directory)에 저장할 수 있다. 분석 결과 파일은 소스 코드에 포함되는 함수들 간의 경로, 소스 코드에 포함되는 함수 및 변수를 설정하거나 읽을 수 있는 경로, 소스 코드에 포함되는 함수들 간의 노드 관계(또는 상속 관계) 등을 시각화한 데이터를 포함할 수 있다. 분석 서버(104)는 소스 코드를 분석 및 시각화하기 위한 오픈 소스 소프트웨어들과 연동하여 분석 작업을 수행할 수 있다. 분석 서버(104)는 오픈 소스 소프트웨어들을 동작시키기 위해 필요한 실행 설정 파일을 자동으로 생성할 수 있다.

[0036] 도 2는 예시적인 실시 예에 따른 분석 서버(104)의 구성을 나타낸 블록도이다.

[0037] 도 2를 참조하면, 분석 서버(104)는 사용자 관리 모듈(111), 분석 관리 모듈(113), 설정 파일 생성 모듈(115), 소스 분석 모듈(117), 및 데이터베이스(119)를 포함할 수 있다.

[0038] 사용자 관리 모듈(111)은 각 클라이언트(102)들에 대해 사용자 등록을 수행할 수 있다. 즉, 사용자 관리 모듈(111)은 각 클라이언트(102)들로부터 해당 사용자의 사용자 식별 정보를 수신하고, 수신한 사용자 식별 정보를 데이터베이스(119)에 저장하여 사용자 등록을 수행할 수 있다. 사용자 관리 모듈(111)은 분석 서버(104)에 접속 시도하는 클라이언트(102)를 대상으로 사용자 인증을 수행할 수 있다. 즉, 사용자 관리 모듈(111)은 분석 서버(104)에 접속 시도하는 클라이언트(102)로부터 사용자 식별 정보를 수신하고, 기 저장된 사용자 식별 정보들 중 상기 수신한 사용자 식별 정보와 일치하는 것이 있는지 확인하여 사용자 인증을 수행할 수 있다. 사용자 인증이 완료된 경우, 사용자 관리 모듈(111)는 해당 클라이언트(102)의 접속을 허용할 수 있다.

[0039] 분석 관리 모듈(113)은 각 클라이언트(102)들로부터 수신한 소스 코드 파일을 관리할 수 있다. 즉, 분석 관리 모듈(113)은 클라이언트(102)로부터 소스 코드 분석 요청이 수신되는 경우, 소스 코드 분석 요청에 포함되는 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 데이터베이스(119)에서 해당 사용자의 디렉토리에 저장할 수 있다. 또한, 분석 관리 모듈(113)은 소스 코드 분석 요청에 포함되는 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보를 설정 파일 생성 모듈(115)로 전달하여 소스 분석 작업에 필요한 실행 설정 파일의 생성을 요청할 수 있다. 또한, 분석 관리 모듈(113)은 소스 코드 파일을 분석한 분석 결과 파일을 데이터베이스(119)에서 해당 사용자의 디렉토리에 저장할 수 있다. 분석 관리 모듈(113)은 분석 결과 파일을 해당 클라이언트(102)로 전송할 수 있다. 이때, 분석 관리 모듈(113)은 분석 결과 파일을 웹 페이지의 형태로 해당 클라이언트(102)에 제공할 수 있다.

[0040] 설정 파일 생성 모듈(115)은 소스 코드 분석 요청에 포함되는 소스 코드 파일의 분석을 위해 필요한 실행 설정 파일을 자동으로 생성할 수 있다. 즉, 설정 파일 생성 모듈(115)은 소스 분석 모듈(117)에서 상기 소스 코드 파일을 분석하기 위한 오픈 소스 소프트웨어를 실행시키기 위해 필요한 실행 설정 파일을 자동으로 생성할 수 있다.

[0041] 구체적으로, 설정 파일 생성 모듈(115)은 소스 코드 분석 요청에 포함되는 해당 소스 코드의 프로그래밍 언어 정보를 통해 해당 소스 코드가 어떤 프로그래밍 언어로 작성되었는지를 확인할 수 있다. 한편, 분석 서버(104)에는 각 프로그래밍 언어 별로 기본적인 설정이 되어있는 기본 설정 파일들이 저장될 수 있다. 여기서, 설정 파일 생성 모듈(115)은 기 저장된 기본 설정 파일들 중 상기 프로그래밍 언어에 대응되는 기본 설정 파일을 추출할 수 있다.

[0042] 프로그래밍 언어 별 기본 설정 파일은 모든 프로그래밍 언어에 공통적으로 적용되는 공통 설정 파일 및 해당 프로그래밍 언어에 특수하게 적용되는 특수 설정 파일을 포함할 수 있다. 표 1은 본 발명의 예시적인 실시 예에 따른 기본 설정 파일 중 모든 프로그래밍 언어에 공통적으로 적용되는 공통 설정 파일 내용을 나타낸 표이다.

표 1

[0043]

설정 옵션	설정 내용	설명
DOXYFILE_ENCODING	EUC-KR	인코딩 형태에 대한 지정
OUTPUT_LANGUAGE	Korean	결과물을 표시할 언어 지정
GENERATE_LATEX	NO	PDF 파일을 생성해낼 것인가
HAVE_DOT	YES	Graphviz가 설치되어 있는가
DOT_PATH	/usr/bin/dot	Graphviz가 설치된 경로는 어디인가
RECURSIVE	YES	지정된 디렉토리의 하위 디렉토리들에 있는 파일들까지 분석을 할 것인가
GENERATE_TREEVIEW	YES	트리 뷰 형태로 분석 결과를 확인할 것인가
SOURCE_BROWSER	YES	소스 코드를 브라우저에서 확인할 것인가
CALL_GRAPH	YES	임의의 함수가 호출하는 함수들에 대하여 그래프로 표시할 것인가
CALLER_GRAPH	YES	임의의 함수를 호출하는 함수들에 대하여 그래프로 표시할 것인가
DOT_NUM_THREADS	32	그래프를 생성하기 위해 필요한 정보인 DOT를 생성하는 작업을 하는 쓰레드의 개수를 몇 개로 지정할 것인가

[0045] 표 1에서 보는 바와 같이, 공통 설정 파일은 인코딩 형태를 어떻게 지정할 것인지, 결과물을 어떤 언어로 표시할 것인지, PDF 파일을 생성할 것인지, Graphviz(즉, 시각화 툴로서 제2 오픈 소스 소프트웨어(117b))가 설치되어 있는지, Graphviz가 설치된 경로는 어디인지, 지정된 디렉토리의 하위 디렉토리에 있는 파일들까지 분석할 것인지, 트리뷰(Tree View) 형태로 분석 결과를 확인할 것인지, 소스 코드를 브라우저에서 확인할 것인지, 임의의 함수가 호출하는 함수들에 대해 그래프로 표시할 것인지, 임의의 함수를 호출하는 함수들에 대해 그래프로 표시할 것인지, 그래프를 생성하기 위해 필요한 정보인 DOT를 생성하는 작업을 수행하는 쓰레드의 개수를 몇 개로 지정할 것인지 등에 대한 설정 내용을 포함할 수 있다.

[0046] 표 2는 본 발명의 예시적인 실시 예에 따른 기본 설정 파일 중 Java 언어에 특수하게 적용되는 특수 설정 파일 내용을 나타낸 표이다. 표 2에서 보는 바와 같이, Java 언어에 특수하게 적용되는 특수 설정 파일은 클래스에 상속되어진 멤버 변수를 표시할 것인지, 어떤 파일들을 분석할 것인지, 객체 지향 언어에 맞게 최적화 할 것인지, 클래스의 프라이빗(Private) 멤버들까지 분석할 것인지 등에 대한 설정 내용을 포함할 수 있다.

표 2

[0047]

설정 옵션	설정 내용	설명
INLINE_INHERITED_MEMB	YES	클래스에 상속되어진 멤버 변수를 표시할 것인가
FILE_PATTERNS	*.java	어떤 파일들을 분석할 것인가
OPTIMIZE_FOR_JAVA	YES	객체 지향 언어에 맞게 최적화 할 것인가
EXTRACT_ALL	YES	클래스의 private멤버들까지도 분석할 것인가

[0049] 표 3은 본 발명의 예시적인 실시 예에 따른 기본 설정 파일 중 C 언어에 특수하게 적용되는 특수 설정 파일 내용을 나타낸 표이다. 표 3에서 보는 바와 같이, C 언어에 특수하게 적용되는 특수 설정 파일은 어떤 파일들을 분석할 것인지, C 언어 분석에 대해 최적화 할 것인지 등에 대한 설정 내용을 포함할 수 있다.

표 3

[0050]

설정 옵션	설정 내용	설명
FILE_PATTERNS	*.c *.h	어떤 파일들을 분석할 것인가
OPTIMIZE_FOR_C	YES	C언어 분석에 대해 최적화 할 것인가

[0052] 표 4는 본 발명의 예시적인 실시 예에 따른 기본 설정 파일 중 Python 언어에 특수하게 적용되는 특수 설정 파일 내용을 나타낸 표이다. 표 4에서 보는 바와 같이, Python 언어에 특수하게 적용되는 특수 설정 파일은 클래스에 상속되어진 멤버 변수를 표시할 것인지, 어떤 파일들을 분석할 것인지, 객체 지향 언어에 맞게 최적화 할 것인지, 클래스의 프라이빗(Private) 멤버들까지 분석할 것인지, 파이썬(Python) 분석을 위한 필터의 경로를 어떻게 지정할 것인지 등에 대한 설정 내용을 포함할 수 있다.

표 4

[0053]

설정 옵션	설정내용	설명
INLINE_INHERITED_MEMB	YES	클래스에 상속되어진 멤버 변수를 표시할 것인가
FILE_PATTERNS	*.py	어떤 파일들을 분석할 것인가
OPTIMIZE_FOR_JAVA	YES	객체 지향 언어에 맞게 최적화 할 것인가
EXTRACT_ALL	YES	클래스의 private멤버들까지도 분석할 것인가
INPUT_FILTER	python ~/doxypy.py	파이썬 분석을 위한 필터의 경로 (파이썬 파일들을 분석하기 위해서는 별도의 플러그 인 설치가 필요하며, 본 예시에서는 ~/doxypy.py 파일 형태로 존재함)

[0055] 표 2 내지 표 4에서는 Java 언어, C 언어, Python 언어의 특수 설정 파일 내용을 각각 기재하였는데, Java 언어에 대한 기본 설정 파일은 표 1에 기재된 공통 설정 파일 및 표 2에 기재된 특수 설정 파일을 포함하며, 이는 데이터베이스(119)의 해당 디렉토리(예를 들어, Config 디렉토리)에 Doxyfile_JAVA라는 파일명으로 저장될 수 있다. 또한, C 언어에 대한 기본 설정 파일은 표 1에 기재된 공통 설정 파일 및 표 3에 기재된 특수 설정 파일을 포함하며, 이는 데이터베이스(119)의 해당 디렉토리에 Doxyfile_C라는 파일명으로 저장될 수 있다. 또한, Python 언어에 대한 기본 설정 파일은 표 1에 기재된 공통 설정 파일 및 표 4에 기재된 특수 설정 파일을 포함하며, 이는 데이터베이스(119)의 해당 디렉토리에 Doxyfile_Python이라는 파일명으로 저장될 수 있다.

[0056] 또한, 설정 파일 생성 모듈(115)은 추출한 기본 설정 파일을 소스 코드 분석 요청에 포함되는 설정 정보(예를 들어, 소스 코드 파일의 이름, 프로젝트 이름, 및 분석 결과 파일의 저장 경로 등)에 따라 수정하여 실행 설정 파일을 생성할 수 있다.

[0057] 예를 들어, 사용자 식별 정보가 koain인 클라이언트(102)가 소스 코드 파일의 이름(예를 들어, Source.zip), 프로젝트 이름(예를 들어, my_project), 프로그래밍 언어 정보(예를 들어, Python 언어) 등을 포함하는 소스 코드 분석 요청을 전송한 경우, 설정 파일 생성 모듈(115)은 기 저장된 기본 설정 파일들 중 Python 언어에 대한 기본 설정 파일을 추출하여 해당 사용자의 디렉토리에 복사할 수 있다. 이때, 설정 파일 생성 모듈(115)은 Config/Doxyfile_Python 경로의 파일을 koain/Doxyfile의 파일명으로 복사할 수 있다. 다음으로, 설정 파일 생성 모듈(115)은 Python 언어에 대한 기본 설정 파일을 소스 코드 분석 요청에 포함된 설정 정보를 기반으로 수정하여 실행 설정 파일을 생성할 수 있다. 이때, 설정 파일 생성 모듈(115)은 Python 언어에 대한 기본 설정 파일 중 공통 설정 파일에 상기 설정 정보를 추가하여 실행 설정 파일을 생성할 수 있다. 표 5는 예시적인 실시 예에서 소스 코드 분석 요청에 포함된 설정 정보가 기본 설정 파일에 추가되는 상태를 나타낸 것이다. 표 5에서 보는 바와 같이, Python 언어에 대한 실행 설정 파일에는 상기 설정 정보를 기반으로 프로젝트의 이름, 결과물이 저장될 경로, 및 분석 대상이 되는 소스 코드 파일이 존재하는 경로 등이 추가될 수 있다.

표 5

[0058]

설정 옵션	설정 내용	설명
PROJECT_NAME	my_project	Project의 이름을 설정
OUTPUT_DIRECTORY	koain/my_project	결과물이 저장될 경로를 지정
INPUT	koain/source	분석 대상이 되는 소스 코드 파일이 존재하는 경로를 지정

[0060] 소스 분석 모듈(117)은 소스 코드 분석 요청에 포함되는 소스 코드 파일을 분석하여 분석 결과 파일을 생성할

수 있다. 소스 분석 모듈(117)은 오픈 소스 소프트웨어와 연동하여 소스 코드 파일을 분석할 수 있다. 소스 분석 모듈(117)은 설정 파일 생성 모듈(115)로부터 실행 설정 파일을 전달받아 오픈 소스 소프트웨어를 실행시킬 수 있다. 예시적인 실시 예에서, 소스 분석 모듈(117)은 제1 오픈 소스 소프트웨어(117a), 제2 오픈 소스 소프트웨어(117b), 및 제3 오픈 소스 소프트웨어(117c)를 포함할 수 있다. 제1 오픈 소스 소프트웨어(117a), 제2 오픈 소스 소프트웨어(117b), 및 제3 오픈 소스 소프트웨어(117c)는 플러그-인(Plug-in) 방식으로 소스 분석 모듈(117)에 탑재될 수 있다.

[0061] 제1 오픈 소스 소프트웨어(117a)는 소스 코드 파일에 포함되는 소스 코드들에 대해 해당 소스 코드를 설명하는 주석을 부가하여 일정 형식(예를 들어, PDF(Portable Document Format) 또는 HTML(Hypertext Markup Language) 형식)의 소스 코드 주석 문서를 생성할 수 있다. 즉, 제1 오픈 소스 소프트웨어(117a)는 소스 코드 파일에 포함되는 소스 코드에 주석 처리를 한 소스 코드 주석 문서를 생성하는 오픈 소스 소프트웨어일 수 있다. 예를 들어, 제1 오픈 소스 소프트웨어(117a)로는 Doxygen이 사용될 수 있으나, 이에 한정되는 것은 아니다.

[0062] 제2 오픈 소스 소프트웨어(117b)는 소스 코드 파일에 포함되는 컴포넌트(예를 들어, 함수, 변수, 클래스 등)들 간의 경로, 컴포넌트를 설정하거나 읽을 수 있는 경로, 및 컴포넌트들 간의 관계도(예를 들어, 노드 관계 또는 상속 관계 등) 중 적어도 하나를 그래프 형태로 시각화 할 수 있는 오픈 소스 소프트웨어일 수 있다. 제2 오픈 소스 소프트웨어(117b)는 제1 오픈 소스 소프트웨어(117a)와 연동해서 동작할 수 있다. 즉, 제2 오픈 소스 소프트웨어(117b)는 제1 오픈 소스 소프트웨어(117a)에서 생성된 소스 코드 주석 문서를 입력받아 시각화된 분석 결과 문서를 생성할 수 있다. 예를 들어, 제2 오픈 소스 소프트웨어(117b)로는 Graphiz가 사용될 수 있으나, 이에 한정되는 것은 아니다.

[0063] 제3 오픈 소스 소프트웨어(117c)는 자국 언어 지원을 위한 오픈 소스 소프트웨어일 수 있다. 즉, 제3 오픈 소스 소프트웨어(117c)는 제1 오픈 소스 소프트웨어(117a)가 자국 언어를 지원하지 않는 경우, 제1 오픈 소스 소프트웨어(117a)가 생성한 소스 코드 주석 문서를 유니코드(Unicode)로 변환하고, 변환된 유니코드를 자국어로 텍스트 변환하는 오픈 소스 소프트웨어일 수 있다. 예를 들어, 제3 오픈 소스 소프트웨어(117c)는 Libiconv가 사용될 수 있으나, 이에 한정되는 것은 아니다.

[0064] 여기서는, 소스 분석 모듈(117)이 오픈 소스 소프트웨어를 이용하여 소스 코드를 분석하는 것으로 설명하였으나, 이에 한정되는 것은 아니며 오픈 소스 소프트웨어 이외의 다른 소프트웨어를 이용하여 소스 코드를 분석할 수도 있음은 물론이다.

[0065] 데이터베이스(119)는 각 사용자 별로 디렉토리가 구분될 수 있다. 데이터베이스(119)는 사용자 별로 사용자 식별 정보를 저장할 수 있다. 데이터베이스(119)는 각 클라이언트(102)들로부터 수신한 소스 코드 파일 및 해당 소스 코드의 프로그래밍 언어 정보를 저장할 수 있다. 데이터베이스(119)는 각 프로그래밍 언어 별로 기본 설정 파일을 저장할 수 있다. 데이터베이스(119)는 소스 분석 모듈(117)이 생성하는 분석 결과 파일을 저장할 수 있다.

[0066] 본 발명의 실시 예에 의하면, 클라우드 기반의 소스 코드 분석 서비스를 제공함으로써, 코드 분석 실행 시 분석 수행 시간을 줄일 수 있으며, 그로 인해 대용량의 코드 분석을 신속하게 수행할 수 있게 된다. 또한, 프로그래밍 언어 별로 코드 분석 시 필요한 실행 설정 파일을 자동으로 생성하여 사용자의 편의성을 향상시킬 수 있게 된다.

[0068] 도 3은 예시적인 실시 예에 따른 소스 코드 분석 방법을 나타낸 흐름도이다. 도시된 흐름도에서는 상기 방법을 복수 개의 단계로 나누어 기재하였으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는 도시되지 않은 하나 이상의 단계가 부가되어 수행될 수 있다. 또한 실시 예에 따라 상기 방법에 도시되지 않은 하나 이상의 단계들이 상기 방법과 함께 수행될 수도 있다. 또한, 상기 방법은 전술한 분석 서버(104)에 의해 수행될 수 있다.

[0069] 도 3을 참조하면, 분석 서버(104)는 클라이언트(102)로부터 소스 코드 분석 요청을 수신한다(S 101). 소스 코드 분석 요청에는 소스 코드 파일, 해당 소스 코드의 프로그래밍 언어 정보, 및 설정 정보가 포함될 수 있다. 설정 정보는 실행 설정 파일을 생성하기 위해 필요한 정보로서, 예를 들어, 소스 코드 파일의 이름, 소스 코드 경로, 및 분석 결과 파일의 저장 경로 등을 포함할 수 있다.

[0070] 다음으로, 분석 서버(104)는 소스 코드 분석 요청에 포함된 해당 소스 코드의 프로그래밍 언어 정보에 기반으로

기 저장된 기본 설정 파일들 중 해당 프로그래밍 언어에 대응하는 기본 설정 파일을 추출한다(S 103).

[0071] 다음으로, 분석 서버(104)는 소스 코드 분석 요청에 포함된 설정 정보를 기반으로 추출한 기본 설정 파일을 수정하여 실행 설정 파일을 생성한다(S 105). 여기서, 실행 설정 파일은 소스 분석 모듈(117)에서 소스 코드 파일을 분석하기 위한 오픈 소스 소프트웨어들의 실행을 위한 파일일 수 있다.

[0072] 다음으로, 분석 서버(104)는 실행 설정 파일을 이용하여 소스 코드 파일을 분석하기 위한 제1 오픈 소스 소프트웨어(117a)를 실행한다(S 107). 제1 오픈 소스 소프트웨어(117a)는 소스 코드 파일에 포함되는 소스 코드에 주석 처리를 한 소스 코드 주석 문서를 생성하는 오픈 소스 소프트웨어일 수 있다.

[0073] 다음으로, 분석 서버(104)는 실행 설정 파일을 이용하여 소스 코드 파일을 시각화하기 위한 제2 오픈 소스 소프트웨어(117b)를 실행한다(S 109). 제2 오픈 소스 소프트웨어(117b)는 소스 코드 파일에 포함되는 컴포넌트(예를 들어, 함수, 변수, 클래스 등)들 간의 경로, 컴포넌트를 설정하거나 읽을 수 있는 경로, 및 컴포넌트들 간의 관계도(예를 들어, 노드 관계 또는 상속 관계 등) 중 적어도 하나를 그래프 형태로 시각화 할 수 있는 오픈 소스 소프트웨어일 수 있다. 제2 오픈 소스 소프트웨어(117b)는 제1 오픈 소스 소프트웨어(117a)에서 생성된 소스 코드 주석 문서를 입력받아 시각화된 분석 결과 문서를 생성할 수 있다.

[0074] 다음으로, 분석 서버(104)는 제1 오픈 소스 소프트웨어(117a)가 자국어를 지원하는지 여부를 확인하여(S 111), 자국어를 지원하지 않는 경우, 제3 오픈 소스 소프트웨어(117c)를 실행하여 제1 오픈 소스 소프트웨어(117a)가 생성한 소스 코드 주석 문서를 유니코드(Unicode)로 변환하고, 변환된 유니코드를 자국어로 텍스트 변환한다(S 113).

[0075] 다음으로, 분석 서버(104)는 소스 코드 파일을 분석한 분석 결과 파일을 클라이언트(102)로 제공한다(S 115). 도 4는 예시적인 실시 예에 따른 분석 서버(104)에서 소스 코드 파일의 분석 결과를 웹 페이지에 표시하는 상태를 나타낸 도면이다.

[0077] 도 5는 예시적인 실시 예들에서 사용되기에 적합한 컴퓨팅 장치를 포함하는 컴퓨팅 환경(10)을 예시하여 설명하기 위한 블록도이다. 도시된 실시 예에서, 각 컴포넌트들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있고, 이하에 기술되지 것 이외에도 추가적인 컴포넌트를 포함할 수 있다.

[0078] 도시된 컴퓨팅 환경(10)은 컴퓨팅 장치(12)를 포함한다. 일 실시 예에서, 컴퓨팅 장치(12)는 클라이언트(예를 들어, 클라이언트(102))일 수 있다. 또한, 컴퓨팅 장치(12)는 소스 코드를 분석하기 위한 장치(예를 들어, 분석 서버(104))일 수 있다.

[0079] 컴퓨팅 장치(12)는 적어도 하나의 프로세서(14), 컴퓨터 판독 가능 저장 매체(16) 및 통신 버스(18)를 포함한다. 프로세서(14)는 컴퓨팅 장치(12)로 하여금 앞서 언급된 예시적인 실시 예에 따라 동작하도록 할 수 있다. 예컨대, 프로세서(14)는 컴퓨터 판독 가능 저장 매체(16)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 상기 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 상기 컴퓨터 실행 가능 명령어는 프로세서(14)에 의해 실행되는 경우 컴퓨팅 장치(12)로 하여금 예시적인 실시 예에 따른 동작들을 수행하도록 구성될 수 있다.

[0080] 컴퓨터 판독 가능 저장 매체(16)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능 저장 매체(16)에 저장된 프로그램(20)은 프로세서(14)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시 예에서, 컴퓨터 판독 가능 저장 매체(16)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 컴퓨팅 장치(12)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0081] 통신 버스(18)는 프로세서(14), 컴퓨터 판독 가능 저장 매체(16)를 포함하여 컴퓨팅 장치(12)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0082] 컴퓨팅 장치(12)는 또한 하나 이상의 입출력 장치(24)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(22) 및 하나 이상의 네트워크 통신 인터페이스(26)를 포함할 수 있다. 입출력 인터페이스(22) 및 네트워크 통신 인터페이스(26)는 통신 버스(18)에 연결된다. 입출력 장치(24)는 입출력 인터페이스(22)를 통해 컴퓨팅 장치(12)의 다른 컴포넌트들에 연결될 수 있다. 예시적인 입출력 장치(24)는 포인팅 장치(마우스 또는 트랙패드

등), 키보드, 터치 입력 장치(터치패드 또는 터치스크린 등), 음성 또는 소리 입력 장치, 다양한 종류의 센서 장치 및/또는 촬영 장치와 같은 입력 장치, 및/또는 디스플레이 장치, 프린터, 스피커 및/또는 네트워크 카드와 같은 출력 장치를 포함할 수 있다. 예시적인 입출력 장치(24)는 컴퓨팅 장치(12)를 구성하는 일 컴포넌트로서 컴퓨팅 장치(12)의 내부에 포함될 수도 있고, 컴퓨팅 장치(12)와는 구별되는 별개의 장치로 컴퓨팅 장치(12)와 연결될 수도 있다.

[0084] 이상에서 본 발명의 대표적인 실시 예들을 상세하게 설명하였으나, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 상술한 실시 예에 대하여 본 발명의 범주에서 벗어나지 않는 한도 내에서 다양한 변형이 가능함을 이해할 것이다. 그러므로 본 발명의 권리범위는 설명된 실시 예에 국한되어 정해져서는 안 되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

[0086] 100 : 소스 코드 분석 시스템

102 : 클라이언트

102a : 유저 인터페이스 모듈

104 : 분석 서버

111 : 사용자 관리 모듈

113 : 분석 관리 모듈

115 : 설정 파일 생성 모듈

117 : 소스 분석 모듈

117a : 제1 오픈 소스 소프트웨어

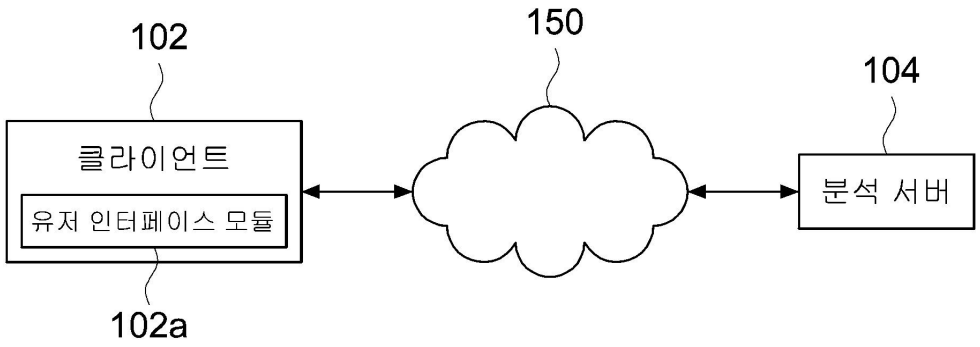
117b : 제2 오픈 소스 소프트웨어

117c : 제3 오픈 소스 소프트웨어

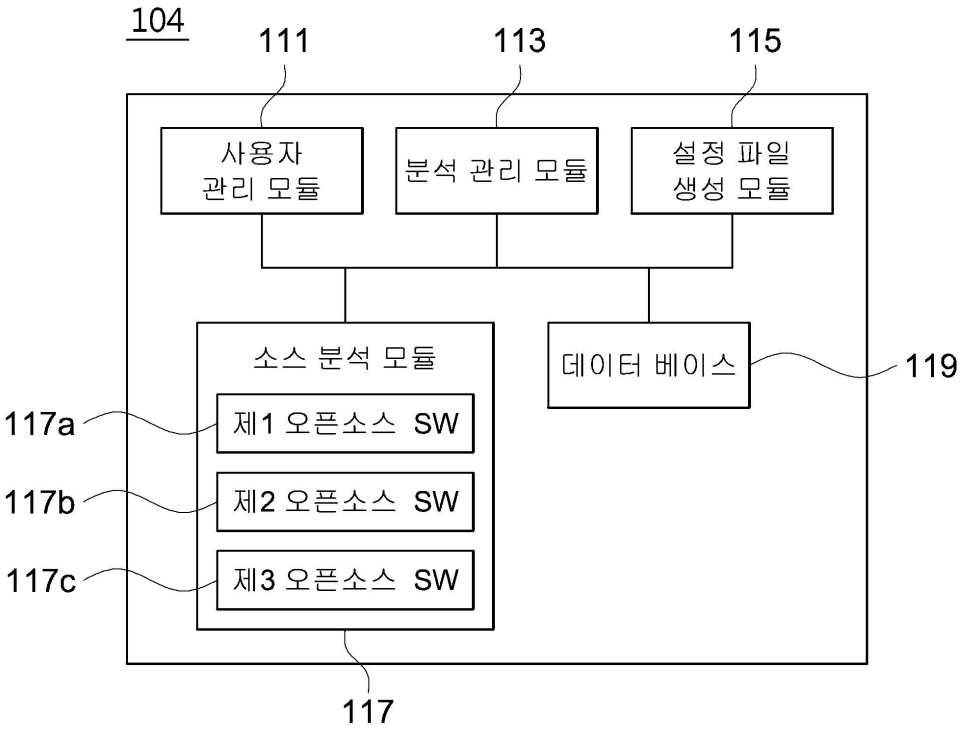
119 : 데이터베이스

도면

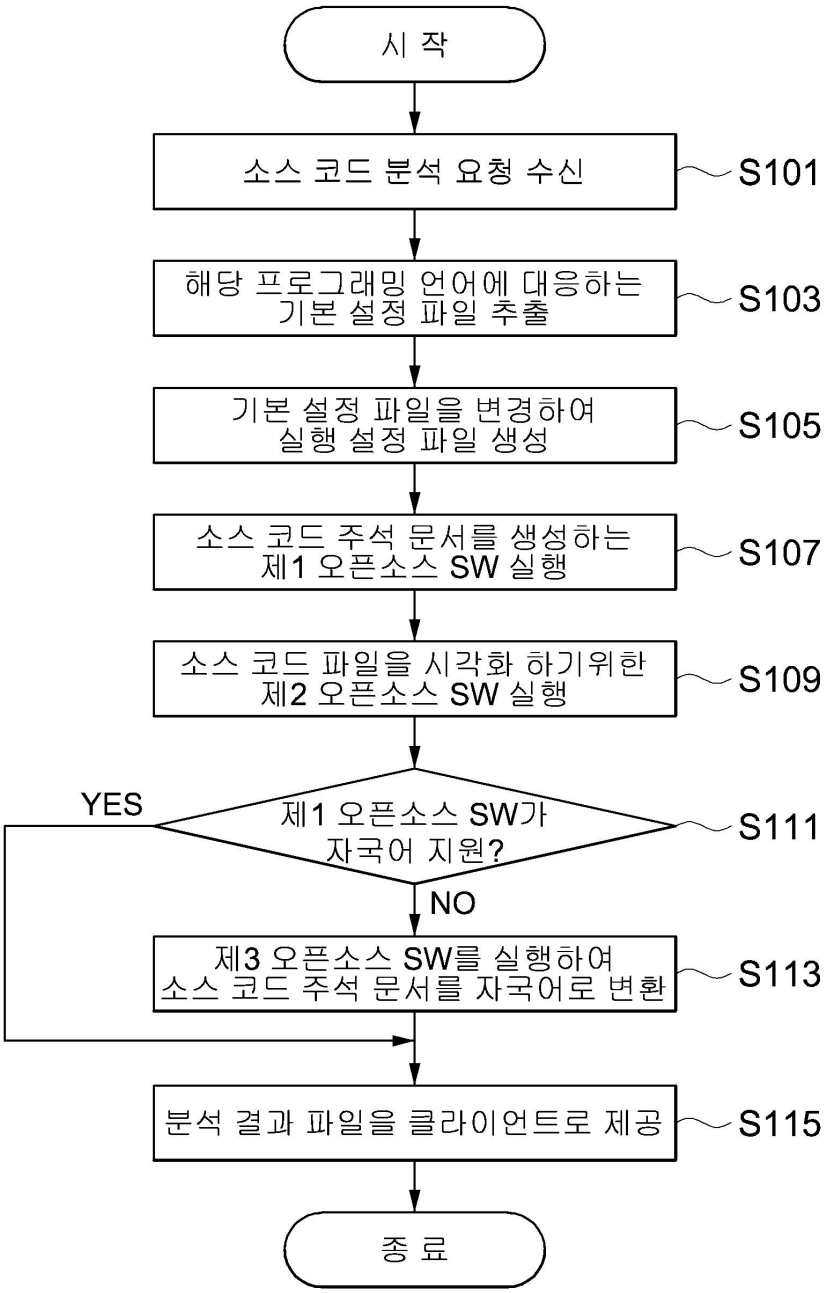
도면1



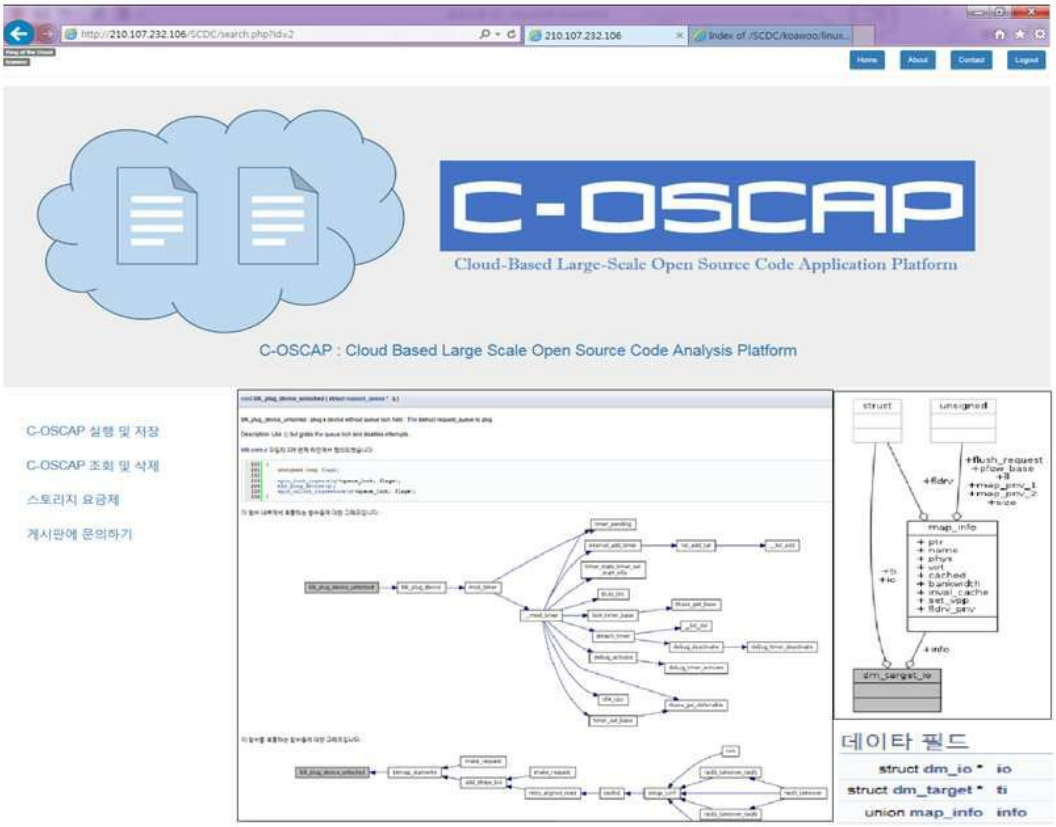
도면2


104
111
113
115
사용자
관리 모듈
분석 관리 모듈
설정 파일
생성 모듈
소스 분석 모듈
데이터 베이스
119
117a
제1 오픈소스 SW
117b
제2 오픈소스 SW
117c
제3 오픈소스 SW
117

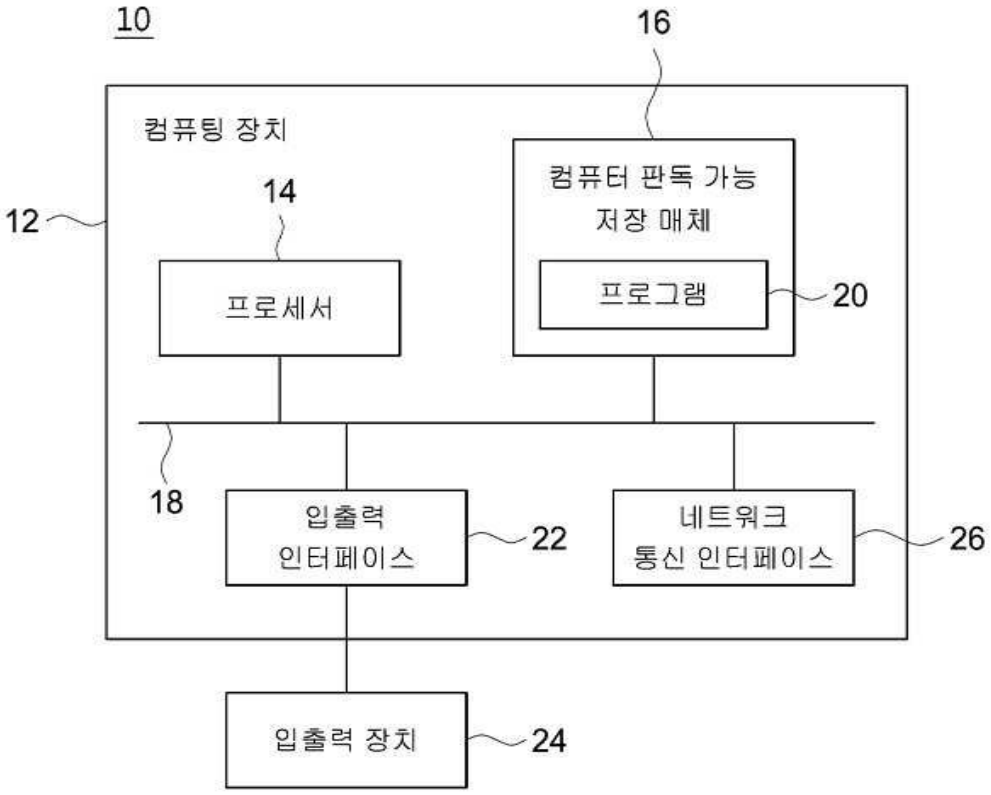
도면3


시 작
소스 코드 분석 요청 수신
S101
해당 프로그래밍 언어에 대응하는 기본 설정 파일 추출
S103
기본 설정 파일을 변경하여 실행 설정 파일 생성
S105
소스 코드 주석 문서를 생성하는 제1 오픈소스 SW 실행
S107
소스 코드 파일을 시각화 하기위한 제2 오픈소스 SW 실행
S109
제1 오픈소스 SW가 자국어 지원?
S111
YES
NO
제3 오픈소스 SW를 실행하여 소스 코드 주석 문서를 자국어로 변환
S113
분석 결과 파일을 클라이언트로 제공
S115
종 료

도면4


C-OSCAP
Cloud-Based Large-Scale Open Source Code Application Platform
C-OSCAP : Cloud Based Large Scale Open Source Code Analysis Platform
C-OSCAP 실행 및 저장
C-OSCAP 조회 및 삭제
스토리지 요금제
게시판에 문의하기
데이타 필드
struct dm_io * io
struct dm_target * ti
union map_info info

도면5


10
16
컴퓨팅 장치
14
12
컴퓨터 판독 가능
저장 매체
프로세서
프로그램
20
18
입출력
인터페이스
22
네트워크
통신 인터페이스
26
입출력 장치
24