



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0034825
(43) 공개일자 2011년04월06일

(51) Int. Cl.

G06F 11/36 (2006.01) G06F 11/26 (2006.01)

(21) 출원번호 10-2009-0092272

(22) 출원일자 2009년09월29일

심사청구일자 2009년09월29일

(71) 출원인

현대자동차주식회사

서울 서초구 양재동 231

이화여자대학교 산학협력단

서울 서대문구 대현동 11-1 이화여자대학교내

기아자동차주식회사

서울특별시 서초구 양재동 231

(72) 발명자

최병주

서울특별시 강남구 압구정동 현대아파트 74동 1305호

서주영

서울특별시 송파구 문정동 올림픽힐리타운 105동 304호

(뒷면에 계속)

(74) 대리인

특허법인태평양

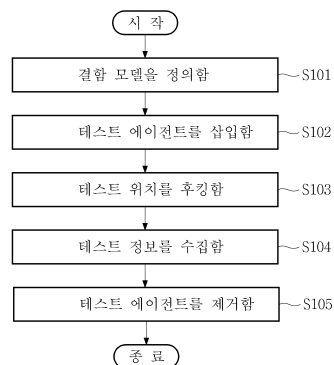
전체 청구항 수 : 총 12 항

(54) 온라인 시스템 테스트 방법

(57) 요약

본 발명은 후킹 기술과 프로파일링 개념을 온라인 시스템 테스트에 적용하여 자원 오버헤드를 최소화한 시스템 테스트와 결함 원인과 위치 식별을 위한 테스트 정보 수집을 동시에 하기 위한 온라인 시스템 테스트 방법과 관련된다. 구체적으로, 본 발명은 결함 모델을 정의하는 단계, 테스트 에이전트를 삽입하는 단계, 테스트 위치를 후킹하는 단계, 테스트 정보를 수집하는 단계, 및 테스트 에이전트를 제거하는 단계를 포함하는 온라인 시스템 테스트 방법을 개시한다.

대표도 - 도1



(72) 발명자

양승완

경기도 군포시 산본동 삼성아파트 1137동 2202호

권해영

경기도 용인시 기흥구 보정동 현대아이파크1차아파트 202동 601호

특허청구의 범위

청구항 1

결함 모델을 정의하는 단계;
 테스트 에이전트를 삽입하는 단계;
 테스트 위치를 후킹하는 단계;
 테스트 정보를 수집하는 단계; 및
 테스트 에이전트를 제거하는 단계를 포함하는 온라인 시스템 테스트 방법.

청구항 2

청구항 1에 있어서,
 상기 결함 모델을 정의하는 단계는
 결함 유형에 따라 상기 테스트 위치를 정의하는 단계;
 상기 결함 유형에 따라 상기 테스트 정보를 정의하는 단계; 및
 상기 테스트 정보를 수집하기 위한 테스트 코드를 생성하는 단계를 포함하는 온라인 시스템 테스트 방법.

청구항 3

청구항 2에 있어서,
 상기 테스트 에이전트를 삽입하는 단계는
 운영 중인 프로세스에 대한 리스트를 구성하는 단계; 및
 상기 리스트를 기초로 상기 테스트 에이전트를 삽입하는 단계를 포함하는 온라인 시스템 테스트 방법.

청구항 4

청구항 3에 있어서,
 상기 테스트 위치를 후킹하는 단계는
 상기 테스트 위치에 따라 메모리 영역을 식별하는 단계;
 상기 식별된 메모리 영역을 상기 테스트 코드로 후킹하는 단계를 포함하는 온라인 시스템 테스트 방법.

청구항 5

청구항 4에 있어서,
 상기 테스트 정보를 수집하는 단계는
 프로그램 실행 정보, 프로세스 컨텍스트 정보 및 시스템 상태 정보를 수집하는 것을 특징으로 하는 온라인 시스템 테스트 방법.

청구항 6

청구항 5에 있어서,
 상기 테스트 에이전트를 제거하는 단계는
 상기 테스트 위치를 복구하는 단계; 및
 상기 테스트 에이전트를 제거하는 단계를 포함하는 온라인 시스템 테스트 방법.

청구항 7

결함 모델을 정의하는 단계;
 테스트 에이전트를 삽입하는 단계;
 테스트 위치에 따라 메모리 영역을 식별하는 단계;
 상기 식별된 메모리 영역을 후킹하는 단계;
 테스트 정보를 수집하는 단계; 및
 테스트 에이전트를 제거하는 단계를 포함하는 온라인 시스템 테스트 방법.

청구항 8

청구항 7에 있어서,
 상기 결함 모델을 정의하는 단계는
 결함 유형에 따라 상기 테스트 위치를 정의하는 단계;
 상기 결함 유형에 따라 상기 테스트 정보를 정의하는 단계; 및
 상기 테스트 정보를 수집하기 위한 테스트 코드를 생성하는 단계를 포함하는 온라인 시스템 테스트 방법.

청구항 9

청구항 8에 있어서,
 상기 식별된 메모리 영역을 후킹하는 단계는
 상기 식별된 메모리 영역을 상기 테스트 코드로 후킹하는 것을 특징으로 하는 온라인 시스템 테스트 방법.

청구항 10

청구항 9에 있어서,
 상기 테스트 위치가 SYSTEM API(Application Program Interface) 함수인 경우 CPS(Current Process Space)의 APISet 영역을 후킹하는 것을 특징으로 하는 온라인 시스템 테스트 방법.

청구항 11

청구항 9에 있어서,
 상기 테스트 위치가 DLL(Dynamic Link Library) EXPORT 함수인 경우 CPS의 IAT(Import Address Table) 영역을 후킹하는 것을 특징으로 하는 온라인 시스템 테스트 방법.

청구항 12

청구항 9에 있어서,
 상기 테스트 위치가 SYSTEM API 또는 DLL EXPORT 함수가 아닌 경우 CPS의 코드 공간 영역을 후킹하는 것을 특징으로 하는 온라인 시스템 테스트 방법.

명세서

발명의 상세한 설명

기술 분야

[0001] 본 발명은 온라인 시스템을 테스트하는 방법과 관련된다.

배경 기술

[0002] 최근 IT가 기존 전통 산업에 접목되는 'IT 융합'이 각광받고 있으며, 의료-IT, 자동차-IT, 항공-IT, 건설-IT와 같이 다양한 산업 분야에서 경쟁력 강화의 해결사로 IT 융합을 추진하고 있다. 일찍이 의료 분야의 IT 접목은

수술 시 의료사고를 최소화하기 위해 시작되었으며, 신속 정확한 영상 정보를 제공하고, 정밀한 수술은 로봇이 대신하는 '원격로봇 영상유도시스템' 연구가 많은 결실을 보이고 있다. 자동차 분야의 경우도 '텔레매틱스 (telematics), 내비게이션 및 위치 기반 서비스 (LBS: Location Based Service), 차량용 인터페이스 (HMI: Human Machine Interface)'를 중심으로 지능형 자동차 개발을 위해 IT 융합을 시도하고 있다.

- [0003] IT 융합 산업에서의 소프트웨어는, 소프트웨어가 독립적으로 동작하는 것보다 융합되는 시스템의 운영환경에서 상호작용(interaction)하는 상황의 동작이 무엇보다 중요해진다. 또한, 소프트웨어 자체만으로 동작을 확인할 수 있는 부분이 상대적으로 제한되기 때문에 테스트를 위한 가상 데이터가 아닌, 시스템이 실제 운영되면서 얻게 되는 실제 데이터(live environment with real data)를 입력으로 한 동작이 매우 중요하며, 온라인 운영 타임(online operation time)에서의 시스템 테스트가 매우 중요한 이슈이다. 특히 온라인 운영 타임에선 오랜 기간 동안 시스템을 운영(런)하는 과정에서 발생하는 잠재 결함이나 시스템 다운을 야기시키는 결함을 발견할 수 있는 테스트가 요구된다.
- [0004] 종래 기술에 따른 시스템 테스트 방법의 하나로 블랙박스 테스트가 있다. 블랙박스 테스트는 시스템의 사용 시나리오를 테스트케이스로 시스템이 요구사항에 맞춰 올바르게 동작하는 지를 확인하는 방식으로 이루어진다.
- [0005] 하지만 블랙박스 테스트는 발견된 결함의 발생 위치와 원인 식별이 어렵다는 문제점이 있다.
- [0006] 대부분의 산업 현장에선 짧은 제품 주기와 시장 선점을 위한 짧은 개발 일정으로 인해 단계적으로 면밀한 테스트를 하기보다는 개발 후반에 강력한 시스템 테스트를 통해 품질을 안정시키는 경향이 있다. 이러한 테스트 경향은 시스템을 구성하는 소프트웨어 각각을 별도로 독립적으로 테스트하려면 연결되는 상대 컴포넌트를 대체하기 위한 장비를 갖추어야 하는 등 테스트 자원과 비용이 많이 필요하기 때문이다.
- [0007] 이렇듯 개발 후반부에 집중된 테스트는 시스템을 구성하는 모든 계층에 걸쳐 결함이 포함되어 있을 수밖에 없으며, 블랙박스 테스트로는 결함의 원인 파악과 위치에 대한 식별이 어려워져 테스트 기간이 늘어남으로써 결국 개발 비용이 증가하는 문제점이 있다.
- [0008] 종래 기술에 따른 시스템 테스트 정보를 얻는 방법으로 디버거를 이용하는 방법이 있다.
- [0009] 하지만 디버거를 이용하는 방법은 실제 시스템이 운용되는 환경에서 테스트를 수행해야 하는 경우, 디버거 사용이 쉽지 않고, 실시간 시스템과 같이 타이밍에 민감한(time-critical) 특성을 갖는 경우, 디버거의 오버헤드 때문에 시스템이 제대로 동작하지 않을 수 있다는 문제점이 있다.
- [0010] 종래 기술에 따른 시스템 테스트 정보를 얻는 다른 방법으로 대상물에 테스트 코드를 삽입(instrument)하는 방법이 있다.
- [0011] 예를 들어, 타이머 루틴에 CPU나 메모리 사용량을 측정하는 코드를 삽입하여 주기적으로 성능을 프로파일링하거나, API 테스트를 위하여 함수 호출 시 파라미터를 모니터링하는 코드를 삽입하는 것을 말한다.
- [0012] 하지만 대상물에 테스트 코드를 삽입하는 방법은 추가 코드로 인한 메모리 공간 오버헤드(space-overhead)와 해당 코드 수행으로 인한 성능 오버헤드(performance-overhead)와 같이 여전히 온라인 운영 타임을 해치는 문제점이 있다.

발명의 내용

해결 하고자하는 과제

- [0013] 상기 문제점을 해결하기 위하여, 본 발명은 운영체제의 프로세스 관리 방식에 후킹 기술을 도입하고, 시스템 상태 정보 수집에 사용되는 프로파일링 기술을 접목하여, 시스템의 실시간 동작을 최대한 보장하고, 시스템에 운영되는 모든 프로그램(프로세스)에 대한 동적 상황을 테스트하며, 결함 원인과 위치 식별을 위한 테스트 정보도 획득하는 새로운 온라인 시스템 테스트 방법을 개시한다.

과제 해결수단

- [0014] 본 발명은 결함 모델을 정의하는 단계; 테스트 에이전트를 삽입하는 단계;
- [0015] 테스트 위치를 후킹하는 단계; 테스트 정보를 수집하는 단계; 및 테스트 에이전트를 제거하는 단계를 포함하는 온라인 시스템 테스트 방법을 개시한다.
- [0016] 추가적으로, 본 발명의 상기 결함 모델을 정의하는 단계는 결함 유형에 따라 테스트 위치를 정의하는 단계; 상

기 결함 유형에 따라 테스트 정보를 정의하는 단계; 및 상기 테스트 정보를 수집하기 위한 테스트 코드를 생성하는 단계를 포함하는 온라인 시스템 테스트 방법을 개시한다.

[0017] 추가적으로, 본 발명의 상기 테스트 에이전트를 삽입하는 단계는 운영 중인 프로세스에 대한 리스트를 구성하는 단계; 및 상기 리스트를 기초로 상기 테스트 에이전트를 삽입하는 단계를 포함하는 온라인 시스템 테스트 방법을 개시한다.

[0018] 추가적으로, 본 발명의 상기 테스트 위치를 후킹하는 단계는 상기 테스트 위치에 따라 메모리 영역을 식별하는 단계; 상기 식별된 메모리 영역을 상기 테스트 코드로 후킹하는 단계를 포함하는 온라인 시스템 테스트 방법을 개시한다.

[0019] 추가적으로, 본 발명의 상기 테스트 정보를 수집하는 단계는 프로그램 실행 정보, 프로세스 컨텍스트 정보 및 시스템 상태 정보를 수집하는 것을 특징으로 한다.

[0020] 추가적으로, 본 발명의 상기 테스트 에이전트를 제거하는 단계는 상기 테스트 위치를 복구하는 단계; 및 상기 테스트 에이전트를 제거하는 단계를 포함하는 온라인 시스템 테스트 방법을 개시한다.

효과

[0021] 본 발명에 따른 온라인 시스템 테스트 방법은 시스템이 부팅되어 이미지가 모두 메모리에 올라간 상태에서 특정 메모리 공간을 후킹하기 때문에 실제 시스템이 동작하는 중에 모든 프로세스와 동적 상황에 대한 테스트가 가능하다는 장점이 있다.

[0022] 추가적으로, 종래 기술에 따른 시스템 테스트 방법은 시스템 내에 운영되는 프로세스가 많더라도, 프로세스 관리를 위한 특정 메모리 공간(CPS: Current Process Space)엔 현재 실행 중인 현행 프로세스와 쓰레드에 대한 핵심 실행 정보를 저장하지만, 본 발명에 따른 온라인 시스템 테스트 방법은 운영체제를 매개로, 모든 프로그램 각각에 대한 개별적인 코드 삽입이 아닌 CPS의 프로그램 컨텍스트 정보를 활용한 후킹 기법을 사용함으로써 테스트로 인한 시스템 오버헤드를 최소화할 수 있다는 장점이 있다.

[0023] 마지막으로, 본 발명에 따른 온라인 시스템 테스트 방법은 시스템의 성능 측정에 사용되던 프로파일링 기술을 테스트에 적용함으로써 '런-타임, 실시간, 자원' 제약이 심한 온라인 시스템 테스트 시에 결함 발견과 복구를 위한 테스트 정보를 취득할 수 있다는 장점이 있다.

발명의 실시를 위한 구체적인 내용

[0024] 본 출원에서 사용되는 용어들의 의미는 다음과 같다.

[0025] 온라인은 시스템의 기능이나 서비스가 실제 운영 중인 상태 혹은 시스템에 대한 요청을 즉시 이용할 수 있는 상태를 의미한다.

[0026] 온라인 시스템 테스트는 시스템이 배포 설치되어 실제 운영되는 환경에서의 시스템 테스트를 의미한다.

[0027] 런-타임은 컴파일 타임과 비교되는 '시스템이 실행하는 동안, 즉 시스템이 시작되어 종료되기까지 동작하는 동적 상황'을 의미한다. 런-타임에서의 활동은 단순히 프로그램을 실행하는 것뿐 만 아니라 이와 연관된 컴포넌트들을 로딩하고 통합하는 동적 연동 상황, 멀티쓰레드 오퍼레이션, 프로세스들 혹은 쓰레드들 사이의 공유 자원 관리와 같은 다양한 동적 행위를 포함한다.

[0028] 실시간은 시스템 혹은 어떤 동작 모드가 외부 프로세스의 요청을 즉시 처리하여 약속된 시간 내에 그 결과를 되돌려 주는 상황을 의미한다.

[0029] 후킹은 특정 목적을 위해 실행 타임에 시스템의 운영과 제어 방식을 가로채어 원하는 방향으로 변경하는 프로그래밍 기법을 의미한다.

[0030] 후킹 기술은 패치(patch)와 같이 이미 배포된 소프트웨어 제품의 기능 개선이나 사소한 버그를 수정하기 위한 업데이트 프로그램을 만드는데 활용되기도 하고, 스파이웨어와 같이 사용자 동의없이 다른 사람의 데이터 정보에 접근하여 수집하거나 수정하는 기술로 악용되기도 한다. 후킹 기술을 활용하면 '운영체제 엿보기'가 가능하기 때문에 종종 운영체제와 연관된 디버깅 목적으로 활용되기도 한다

[0031] 후킹은 코드 삽입 기술(code instrumentation)의 일종으로, 코드가 삽입되는 시점에 따라 정적 방식과 동적 방식으로 분류한다.

- [0032] 정적 방식은 프로그램이 런하기 전, 즉 컴파일 타임에 소스 코드 혹은 바이너리 코드에 정해진 테스트 코드를 추가하는 방식으로 원본 코드를 변형하는 방안이다. 그렇기 때문에, 정적으로 테스트 코드가 삽입된 프로그램은 런-타임에 항상 고정된 방식으로 테스트가 수행되고, 메모리, CPU의 자원 오버헤드가 시스템 라이프 사이클 동안 고정적으로 소비된다. 또한, DLL 연동과 같은 동적 상황을 고려하기 쉽지 않기 때문에 온라인 테스트에 적용하기엔 부족한 방안이다.
- [0033] 동적 방식은 테스트 코드를 런-타임에 삽입하는 방식이다. 정적 방안과 비교하여, 런-타임에 파악되는 동적 상황도 테스트할 수 있다는 점에서 온라인 테스트에 더 적합한 방안이라 할 수 있다.
- [0034] 본 발명에 따른 온라인 시스템 테스트 방법은 동적 방식의 후킹을 의미하며, 최종 배포 코드를 대상으로 하기 위해 소스코드가 아닌 시스템에 탑재되어 실제 운영되는 바이너리 코드를 후킹 대상으로 한다.
- [0035] 또한, 온라인 시스템 테스트는 시스템 내에서 운영되는 특정 하나의 프로그램에 대한 테스트가 아닌 시스템에서 동작하는 모든 프로그램에 대한 테스트를 의미하기 때문에, 특정 프로그램에 대한 후킹이 아닌, 시스템에 동작하는 모든 프로그램들을 관리하고 운영하는 운영체제를 매개로 한 OS-후킹 개념을 적용한다.
- [0036] 프로파일링은 런-타임에 프로그램 행위나 시스템 상태를 측정하는 성능 분석 기법을 의미한다.
- [0037] 프로파일링은 시스템을 최적화시키기 위한 것이므로, 시스템의 성능을 측정하여 성능 병목 구간을 찾는 것이 중요하다. 본 발명은 테스트를 통해 발견된 결함에 대해서 문제가 시스템의 어느 프로그램에서 야기된 것인지, 프로그램 내의 어떤 부분과 연관이 되는지, 결함의 원인이 무엇인지를 밝히기 위한 테스트 정보 수집에 프로파일링을 활용한다.
- [0038] 이하에서는 첨부한 도면을 참고하여 본 발명의 실시예를 상세하게 설명한다.
- [0039] 도 1은 본 발명에 따른 온라인 시스템 테스트 방법의 순서도이다.
- [0040] 도 1을 참고하면, 본 발명에 따른 온라인 시스템 테스트 방법을 시작한다.
- [0041] 본 발명에 따른 온라인 시스템 테스트 방법은 먼저 결함 모델을 정의한다(S101).
- [0042] 결함 모델을 정의한다는 것은 결함에 따라 테스트되어야 할 위치와 결함 발견과 복구를 위한 테스트 정보를 정의한다는 의미이다.
- [0043] 도 2는 본 발명에 따른 온라인 시스템 테스트 방법에서 결함 모델을 정의하는 구체적인 과정을 나타내는 순서도이다.
- [0044] 도 2를 참고하면, 본 발명에 따른 온라인 시스템 테스트 방법은 결함 유형에 따라 테스트 위치를 정의한다(S1011).
- [0045] 다음으로, 본 발명에 따른 온라인 시스템 테스트 방법은 결함 유형에 따라 테스트 정보를 정의한다(S1012).
- [0046] 그리고나서, 본 발명에 따른 온라인 시스템 테스트 방법은 정의된 테스트 위치에 대한 원본 코드 실행과 테스트 정보 수집을 위한 테스트 코드를 생성한다(S1013).
- [0047] 본 발명에 따른 온라인 시스템 테스트 방법의 결함 모델을 정의하는 과정을 종료한다.
- [0048] 결함 모델을 정의하고 나면, 테스트 에이전트를 삽입한다(S102).
- [0049] 테스트 에이전트를 삽입한다는 것은 테스트 위치 후킹과 테스트 정보 수집 역할을 수행하는 테스트 에이전트를 온라인에 동작하는 모든 프로세스에 삽입(침투)한다는 것을 의미한다.
- [0050] 도 3은 본 발명에 따른 온라인 시스템 테스트 방법에서 테스트 에이전트를 삽입하는 구체적인 과정을 나타내는 순서도이다.
- [0051] 도 3을 참고하면, 본 발명에 따른 온라인 시스템 테스트 방법은 시스템에 운영 중인 프로세스에 대한 리스트를 구성한다(S1021).
- [0052] 다음으로, 본 발명에 따른 온라인 시스템 테스트 방법은 구성된 프로세스 리스트 정보를 근거로 테스트 에이전트를 삽입한다(S1022).
- [0053] 본 발명에 따른 온라인 시스템 테스트 방법의 테스트 에이전트를 삽입하는 과정을 종료한다.

- [0054] 테스트 에이전트를 삽입하고 나면, 본 발명에 따른 온라인 시스템 테스트 방법은 테스트 위치를 후킹한다(S103).
- [0055] 테스트 위치를 후킹한다는 것은 삽입된 테스트 에이전트가 테스트 위치로 정의된 곳을 지나갈 때, 원본 코드의 동작을 포함한 테스트 코드가 실행되도록 OS의 CPS (Current Process Space)를 후킹하는 것을 의미한다.
- [0056] 도 4는 본 발명에 따른 온라인 시스템 테스트 방법에서 테스트 위치를 후킹하는 구체적인 과정을 나타내는 순서도이다.
- [0057] 도 4를 참고하면, 본 발명에 따른 온라인 시스템 테스트 방법은 테스트 위치 별로 CPS에서 후킹할 메모리 영역을 식별한다(S1031).
- [0058] 메모리 영역을 식별한 결과, i) 테스트 위치가 SYSTEM API(Application Process Interface) 함수의 경우, CPS의 APISet 을 후킹 영역으로 식별한다.
- [0059] ii) 테스트 위치가 DLL(Dynamic Linking Library) EXPORT 함수의 경우, CPS의 IAT (Import Address Table)을 후킹 영역으로 식별한다.
- [0060] iii) 그 외의 경우, CPS의 코드 공간을 후킹 영역으로 식별한다.
- [0061] 다음으로, 본 발명에 따른 온라인 시스템 테스트 방법은 식별된 후킹 영역을 테스트 코드로 후킹한다(S1032).
- [0062] 본 발명에 따른 온라인 시스템 테스트 방법의 테스트 위치를 후킹하는 과정을 종료한다.
- [0063] 테스트 위치를 후킹하고 나면, 본 발명에 따른 온라인 시스템 테스트 방법은 테스트 정보를 수집한다(S104).
- [0064] 테스트 정보를 수집한다는 것은 시스템이 테스트 목적으로 후킹된 위치를 실행할 때, 테스트를 위한 정보를 수집한다는 것을 의미한다.
- [0065] 도 5는 본 발명에 따른 온라인 시스템 테스트 방법에서 테스트 정보를 수집하는 구체적인 과정을 나타내는 순서도이다.
- [0066] 도 5를 참고하면, 본 발명에 따른 온라인 시스템 테스트 방법은 결함 원인 식별에 필요한 프로그램 실행 정보를 수집한다(S1041).
- [0067] 다음으로, 본 발명에 따른 온라인 시스템 테스트 방법은 결함 위치 식별에 필요한 프로세스 컨텍스트 정보를 수집한다(S1042).
- [0068] 그리고나서, 본 발명에 따른 온라인 시스템 테스트 방법은 성능 분석에 필요한 시스템 상태 정보를 수집한다(S1043).
- [0069] 본 발명에 따른 온라인 시스템 테스트 방법의 테스트 정보를 수집하는 과정을 종료한다.
- [0070] 테스트 정보를 수집하고 나면, 본 발명에 따른 온라인 시스템 테스트 방법은 테스트 에이전트를 제거한다(S105).
- [0071] 테스트 에이전트를 제거한다는 것은 테스트 목적으로 후킹된 위치를 원본 코드로 되돌리고, 테스트 에이전트를 제거한다는 것을 의미한다.
- [0072] 도 6은 본 발명에 따른 온라인 시스템 테스트 방법에서 테스트 에이전트를 제거하는 구체적인 과정을 나타내는 순서도이다.
- [0073] 도 6을 참고하면, 본 발명에 따른 온라인 시스템 테스트 방법은 테스트 코드로 후킹되었던 테스트 위치들을 원본 코드로 복구(unhook)한다(S1051).
- [0074] 다음으로, 본 발명에 따른 온라인 시스템 테스트 방법은 테스트 에이전트가 삽입되었던 프로세스들로부터 테스트 에이전트를 제거한다(S1052).
- [0075] 본 발명에 따른 온라인 시스템 테스트 방법의 테스트 에이전트를 제거하는 과정을 종료한다.

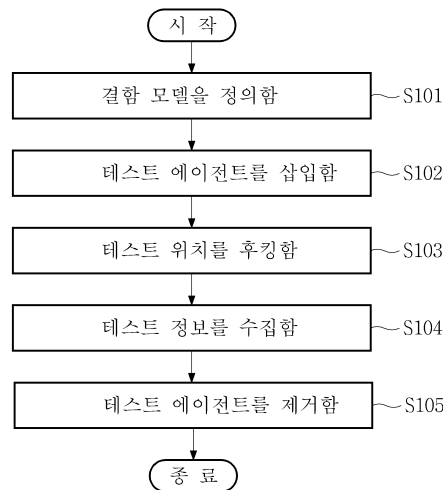
도면의 간단한 설명

- [0076] 도 1은 본 발명에 따른 온라인 시스템 테스트 방법의 순서도이다.

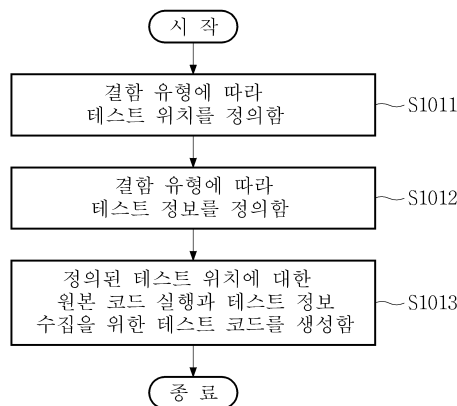
- [0077] 도 2는 본 발명에 따른 온라인 시스템 테스트 방법에서 결함 모델을 정의하는 구체적인 과정을 나타내는 순서도이다.
- [0078] 도 3은 본 발명에 따른 온라인 시스템 테스트 방법에서 테스트 에이전트를 삽입하는 구체적인 과정을 나타내는 순서도이다.
- [0079] 도 4는 본 발명에 따른 온라인 시스템 테스트 방법에서 테스트 위치를 후킹하는 구체적인 과정을 나타내는 순서도이다.
- [0080] 도 5는 본 발명에 따른 온라인 시스템 테스트 방법에서 테스트 정보를 수집하는 구체적인 과정을 나타내는 순서도이다.
- [0081] 도 6은 본 발명에 따른 온라인 시스템 테스트 방법에서 테스트 에이전트를 제거하는 구체적인 과정을 나타내는 순서도이다.

도면

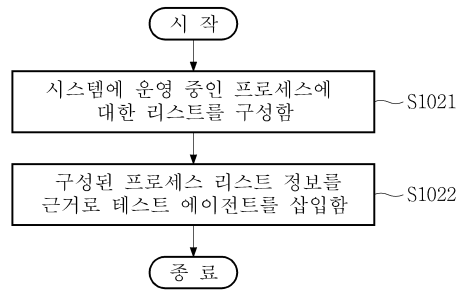
도면1



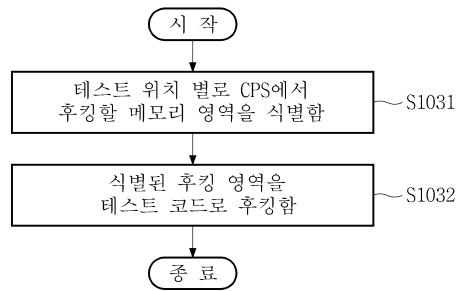
도면2



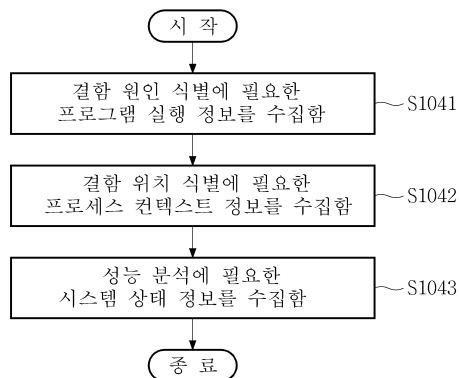
도면3



도면4



도면5



도면6

