(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년11월27일
(11) 등록번호 10-2184071
(24) 등록일자 2020년11월23일

(51) 국제특허분류(Int. Cl.)
G01N 35/00 (2006.01) *G01N 1/28* (2006.01)
G01N 35/08 (2006.01) *G06F 9/06* (2018.01)
G06F 9/30 (2018.01)
(52) CPC특허분류
G01N 35/0092 (2013.01)
G01N 1/28 (2013.01)
(21) 출원번호 10-2015-7011801
(22) 출원일자(국제) 2013년11월07일
심사청구일자 2018년08월28일
(85) 번역문제출일자 2015년05월06일
(65) 공개번호 10-2015-0082284
(43) 공개일자 2015년07월15일
(86) 국제출원번호 PCT/US2013/068886
(87) 국제공개번호 WO 2014/074684
국제공개일자 2014년05월15일
(30) 우선권주장
61/723,736 2012년11월07일 미국(US)
(56) 선행기술조사문헌
EP2450711 A1
(뒷면에 계속)
전체 청구항 수 : 총 10 항

(73) 특허권자
베크만 컬터, 인코포레이티드
미국 92821 캘리포니아주 브리어 사우쓰 크래머 블러바드 250
(72) 발명자
에버하르트 미하엘
독일 80638 뮌헨 필라르슈트라쎄 5
마티네즈 찰스
미국 92887 캘리포니아주 요바 린다 바이아 타라고나 20335
(뒷면에 계속)
(74) 대리인
양영준, 김윤기

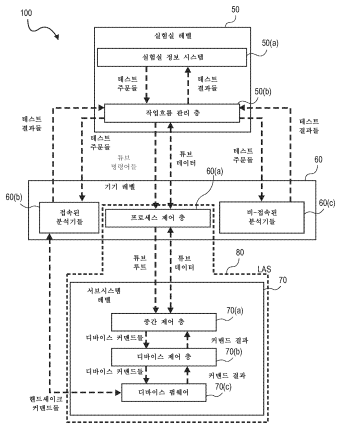
심사관 : 김동원

(54) 발명의 명칭 **자동화된 샘플 처리 시스템**

(57) 요 약

방법은 샘플 컨테이너 내의 샘플에 관한 명령어 데이터를 수신하는 단계를 포함한다. 방법은 작업흐름 관리 층을 이용하는 적어도 하나의 프로세서에 의해 샘플에 대한 프로세스 계획을 생성하는 단계, 및 프로세스 계획을 프로세스 제어 층에 제공하는 단계를 포함한다. 프로세스 계획은 복수의 가능한 루트들을 포함한다. 방법은 또한 프로세스 제어 층을 이용하는 적어도 하나의 프로세서에 의해 프로세스 계획에서의 복수의 가능한 루트들 중의 최적화된 루트를 선택하는 단계, 및 최적화된 루트를 미들웨어 제어 층에 제공하는 단계를 포함한다. 적어도 하나의 프로세서 및 미들웨어 제어 층은 수송 시스템으로 하여금 선택된 루트를 따라 진행하게 하도록 동작가능하다.

대 표 도 - 도1b



(52) CPC특허분류
G01N 33/53 (2018.05)
G01N 35/08 (2013.01)
G06F 9/06 (2013.01)
G06F 9/30 (2013.01)
G01N 2035/00495 (2013.01)
G01N 2035/0094 (2013.01)

(72) 발명자
페인 케슬린
미국 46077 인디애나주 자이온스빌 체스트넛 이글 드라이브 6075
벨테 크리스토프
독일 89250 젠덴 에리히-리팅하우스-슈트라쎄 2/1
비데만 베른트
독일 86415 메링 메링거첼러 슈트라쎄 15

(56) 선행기술조사문헌
US6581012 A
US6721615 A
US20120109531 A1*
JP2011242154 A
JP2003506773 A
*는 심사관에 의하여 인용된 문헌

명 세 서

청구범위

청구항 1

자동화된 샘플 처리 방법으로서,

제1 컴퓨터 판독가능 매체 상의 작업흐름 관리 층을 이용하는 제1 프로세서에 의해, 샘플 컨테이너 내의 샘플에 대한 프로세스 계획을 생성하는 단계;

상기 프로세스 계획을 제2 컴퓨터 판독가능 매체 상의 프로세스 제어 층에 제공하는 단계;

상기 프로세스 제어 층을 이용하는 제2 프로세서에 의해, 상기 프로세스 계획과 일치하는 최적화된 루트를 판정하는 단계;

상기 프로세스 제어 층에 의해, 상기 제2 컴퓨터 판독가능 매체 상의 중간 제어 층에 적어도 하나의 루트 레그를 제공하는 단계;

상기 중간 제어 층에 의해, 상기 적어도 하나의 루트 레그를 더 최적화하는 단계; 및

상기 최적화된 루트를 이용하여 상기 샘플을 처리하는 단계를 포함하고,

상기 프로세스 계획은, 상기 샘플 상에서 작동될 필요가 있는 프로세스들에 관한 정보 및 상기 샘플을 처리하는데 이용되는 분석기들의 상태에 관한 정보를 포함하고,

상기 최적화된 루트는 복수의 상이한 샘플 처리 서브시스템들을 포함하고,

상기 적어도 하나의 루트 레그는, 단일 샘플 처리 서브시스템을 포함하는 상기 최적화된 루트의 일부이거나, 상기 모든 복수의 상이한 샘플 처리 서브시스템들은 아닌 두 개 이상의 샘플 처리 서브시스템들 사이의 경로인, 자동화된 샘플 처리 방법.

청구항 2

제1항에 있어서,

상기 상이한 샘플 처리 서브시스템들은 분석기, 분취기, 로봇, 입력 스테이션, 출력 스테이션, 디캡퍼, 리캡퍼, 그리퍼 유닛, 튜브 검사 유닛(TIU), LLD, 및 원심분리기로 이루어진 군으로부터 선택되는, 자동화된 샘플 처리 방법.

청구항 3

제1항 또는 제2항에 있어서,

실험실 정보 시스템을 통해 명령어 데이터를 수신하는 단계, 및 상기 명령어 데이터를 상기 작업흐름 관리 층에 제공하는 단계를 더 포함하고, 상기 작업흐름 관리 층은 적어도 두 개의 서브층들을 포함하는, 자동화된 샘플 처리 방법.

청구항 4

제1항 또는 제2항에 있어서,

상기 중간 제어 층에 의해 서브어셈블리 컨테이너에 명령어들을 제공하는 단계를 더 포함하고, 상기 서브어셈블리 컨테이너는 상기 서브어셈블리 컨테이너 내에 배치되는 다수의 서브어셈블리들을 제어하고, 서브어셈블리는 샘플 처리 서브시스템의 일부인, 자동화된 샘플 처리 방법.

청구항 5

제1항 또는 제2항에 있어서,

상기 제2 프로세서 및 상기 중간 제어 층에 의해, 디바이스 제어 층에 디바이스 커맨드들을 제공하는 단계를 더

포함하는, 자동화된 샘플 처리 방법.

청구항 6

제1항 또는 제2항에 있어서,

상기 작업흐름 관리 층에 의해, 상기 프로세스 계획을 생성하는 데 이용되는 정보를 하나 이상의 분석기들로부터 수신하는 단계를 더 포함하는, 자동화된 샘플 처리 방법.

청구항 7

자동화된 샘플 처리 시스템으로서,

작업흐름 관리 층을 포함하는 제1 컴퓨터 판독가능 매체 및 제1 프로세서를 포함하는 작업흐름 관리 제어기; 및

상기 작업흐름 관리 제어기에 연결되고 실험실 자동화 시스템의 일부인 LAS 제어기

를 포함하고,

상기 작업흐름 관리 층은, 샘플 컨테이너 내의 샘플에 대한 프로세스 계획을 생성하고 상기 프로세스 계획을 프로세스 제어 층에 제공하도록 동작가능하고, 상기 프로세스 계획은, 상기 샘플 상에서 작동될 필요가 있는 프로세스들에 관한 정보 및 상기 샘플을 처리하는 데 이용되는 분석기들의 상태에 관한 정보를 포함하고,

상기 LAS 제어기는, 상기 프로세스 제어 층 및 중간 제어 층을 포함하는 제2 컴퓨터 판독가능 매체 및 제2 프로세서를 포함하고,

상기 프로세스 제어 층은,

상기 프로세스 계획과 일치하는 최적화된 루트를 판정하고,

상기 중간 제어 층에 적어도 하나의 루트 레그를 제공하도록

동작가능하고,

상기 최적화된 루트는 복수의 상이한 샘플 처리 서브시스템들을 포함하고,

상기 적어도 하나의 루트 레그는, 단일 샘플 처리 서브시스템을 포함하는 상기 최적화된 루트의 일부이거나, 상기 모든 복수의 상이한 샘플 처리 서브시스템들은 아닌 두 개 이상의 샘플 처리 서브시스템들 사이의 경로이고,

상기 중간 제어 층은 상기 적어도 하나의 루트 레그를 더 최적화하도록 동작가능한, 자동화된 샘플 처리 시스템.

청구항 8

제7항에 있어서,

상기 중간 제어 층은 디바이스 제어 층에 디바이스 커맨드들을 제공하도록 더 동작가능한, 자동화된 샘플 처리 시스템.

청구항 9

제7항 또는 제8항에 있어서,

상기 LAS 제어기는 제1 LAS 제어기이고, 상기 실험실 자동화 시스템은 제1 실험실 자동화 시스템이고,

상기 자동화된 샘플 처리 시스템은, 상기 작업흐름 관리 제어기에 연결된 제2 실험실 자동화 시스템에 포함되는 제2 LAS 제어기를 포함하는, 자동화된 샘플 처리 시스템.

청구항 10

제7항 또는 제8항에 있어서,

상기 LAS 제어기 및 상기 작업흐름 관리 제어기는 서로 독립적으로 동작하도록 구성되는, 자동화된 샘플 처리 시스템.

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

발명의 설명

기 술 분 야

[0001] 관련 출원과의 상호참조

[0002] 본 출원은 2012년 11월 7일자로 출원되었고 발명의 명칭이 "자동화된 샘플 처리 시스템(Automated Sample Processing System)"인 미국 가특허 출원 제61/723,736호에 관한 것이다. 이러한 출원은 모든 목적들을 위해 전체적으로 본 명세서에 참고로 포함된다.

배 경 기 술

[0003] 실험실 시스템들을 위한 소프트웨어 구성들은 알려져 있다.

[0004] 한 가지 개시된 분석 시스템이 EP 2 450 711A1호에 기술되어 있다. 개시된 시스템은, EP 2 450 711A1호의 도 2에서 알 수 있는 바와 같이, 서로 상호작용하여 분석 시스템 내의 샘플 처리 작업흐름을 최적화하는 기기 관리자 및 샘플 작업흐름 관리자를 포함한다.

[0005] 기기 관리자는 테스트 주문들을 사용자 또는 LIS(laboratory information system, 실험실 정보 시스템) 중 어느 한쪽으로부터 수신하고, 커맨드들을 접속된 분석기들, 또는 수송 유닛과 같은 다른 접속된 자동 디바이스들로 전송한다. 샘플 작업흐름 관리자는 프로세스 정보를 접속된 분석기 유닛으로부터 수신하고, 사전구성된 처리 루트들을 기기 관리자로 전송하여 정보를 처리하게 한다.

[0006] 다른 개시된 분석 시스템은 미국 특허 제6,581,012호에 기술되어 있다. 이러한 특허에 기술된 임상 실험실 시스템에 대한 개시된 소프트웨어 아키텍처는, 미국 특허 제6,581,012호의 도2에 개략적으로 도시된, 테스트 요청들로부터 테스트 결과들까지의 작업흐름을 다루기 위한 세 개의 층들을 포함한다. 샘플 스케줄링을 기술하는 다른 참조문헌이 미국 특허 제6,721,615호이다.

[0007] 사용자 인터페이스 층에서, 사용자는 작업흐름 자동화 층으로 포워드(forward)되는 테스트 주문들을 입력한다. 작업흐름 자동화 층은 테스트 명령어들을 테스트 주문들로부터 생성하고, 이들을 물리적 요소 층으로 포워드하여 요청된 테스트를 접속된 분석기에 의해 행하게 한다. 분석기는 대응하는 SPM(specimen processing module, 검체 처리 모듈) 대상물을 통해 테스트 결과를 반환하며, 작업흐름 자동화 층 내에서 결과가 검증되고, 유효하다면, 사용자에 대한 광학적 표현을 위한 사용자 인터페이스 층으로 포워드된다.

[0008] 그러한 시스템들이 유용할 수 있지만, 보다 양호한 샘플 처리 최적화를 제공할 수 있는 시스템들이 계속해서 필요하다. 예를 들어, 샘플 컨테이너들을 위해 기기 관리자들에 의해 생성된 프로세스 계획들은 많은 상이한 잠재적 루트들을 가질 수 있다. 샘플 컨테이너에 의해 취해진 루트는 최적이 아닐 수 있어서, 실험실 시스템의 전체적인 효율을 감소시킬 수 있다. 전체적인 효율을 개선하는 것이 바람직한데, 그 이유는 흔히 실험실 시스템 상에서의 처리 용량보다 더 많은 처리할 샘플들이 있기 때문이다.

[0009] 본 발명의 실시 형태들은 이들 및 다른 문제들을 다룬다.

발명의 내용

[0010] 본 발명의 실시 형태들은 진단 실험실 환경에서, 그리고 특히 샘플 수송 시스템에 의해 접속된 다수의 진단 분석기들을 갖는 자동화된 진단 실험실 환경에서의 개선된 샘플 처리 시스템들에 관한 것이다. 일반적으로, 진단 실험실 환경은 환자 샘플들에 대한 개별 진단 테스트들을 수행하는 다수의 분석기들 및 그러한 테스트들을 위해 그러한 환자 샘플들을 준비하는 시스템을 포함한다. 자동화된 진단 실험실은 샘플 준비 시스템, 다수의 분석기들, 및 샘플 준비 시스템과 개별 분석기들 사이에서 환자 샘플들을 수송하기 위한 시스템을 포함한다. 본 발명은 샘플 준비 시스템의 동작을 제어하는 프로세스 제어 아키텍처, 및 그러한 시스템이 환자 샘플들을 선택하고 준비하여 분석기들로 수송하는 방법에 관한 것이다.

[0011] 본 발명의 실시 형태들에서, 작업흐름 관리 층(WML)이 샘플에 대한 프로세스 계획을 생성할 수 있다. 프로세스 계획은 샘플 상에서 작동될 필요가 있는 프로세스들(예컨대, 면역검정 테스트, 화학적 테스트 등)에 관한 정보, 시스템 상의 분석기들의 상태(예컨대, 이용가능성, 시약 상태 등)에 관한 정보, 및 분석기들 상에서 다운스트림 처리의 준비로 행해질 필요가 있는 샘플의 준비를 위한 특정 단계들을 포함할 수 있다. 프로세스 계획은 WML에 독립적으로 동작할 수 있는 프로세스 제어 층(PCL)에 제공될 수 있다. 적어도 하나의 프로세서와 함께 작업하는 PCL은 처리되기를 기다리고 있는 모든 샘플들을 재검토할 수 있고, 이들을 평가할 수 있다. 샘플들이 픽업되는 순서는 (예컨대, 분석기들 또는 전체 프로세스를 늦추는 기타 처리 디바이스들에서의 샘플들의 임의의 백업이 있는 경우에) 샘플들을 처리하는 데 이용될 분석기들(또는 분석기들과 같은 다른 처리 디바이스들)의 유형들 및 상태들, 및 샘플들 상에서 수행될 테스트들에 의존할 수 있다.

[0012] WML은 단일 층일 수 있거나, 또는 일 실시 형태에서 두 개의 개별 층들일 수 있다. 이러한 실시 형태에서, 제1 층은 각각의 샘플을 전달하도록 기본 분석기 목록을 개발하기 위해 그러한 샘플에 대해 수행될 테스트들의 유형들에 대한 명령어들을 수신한다. 제2 층은 루트를 얻기 위해 WML의 제1 층과 통신하고, 개별 분석기들이 요구된 테스트들을 수행할 수 있는지, 그러한 테스트를 위한 충분한 공급물들 및 시약들을 소유하는지, 그리고 달리 동작가능한지 여부를 확인하기 위해 그러한 분석기들과 통신한다. 이러한 정보는, 루트와 함께, PCL로 송신된다. PCL은 다양한 규칙들에 의존하는 처리를 위한 적절한 환자 샘플을 선택하는 데 그러한 정보를 이용하여, 실험실을 통한 환자 샘플의 유동을 최대화한다.

[0013] 일부 실시 형태들에서, 먼저 처리되는 샘플은 이들 소정 규칙들에 의존할 것이다. 예를 들어, 하나의 규칙은 STAT(짧은 턴어라운드 시간 또는 높은 처리 우선순위를 갖는 샘플 튜브들) 샘플들만이 먼저 픽업되지만 그것이 임의의 다른 비-STAT 샘플보다 더 빨리 처리될 수 있는 경우에만 픽업되는 것일 수 있다는 것이다. 시스템에서 각각의 분석기에 선행하는 입력 버퍼는 PCL에 의해 분석되어, 더 많은 샘플들을 입력 버퍼에 추가하는 것이 시스템을 느리게 하는 지점까지 입력 버퍼가 풀(full) 상태인지를 판정하게 할 수 있다. 예를 들어, STAT 샘플 및 비-STAT 샘플을 포함하는 두 개의 샘플들이 실험실 자동화 시스템(laboratory automation system)에 의해 처리될 필요가 있을 수 있다. STAT 샘플은 면역검정 분석기 A를 통과할 필요가 있을 수 있고, PCL은 면역검정 분석기 A에 의해 처리되기를 기다리는 다른 샘플들의 대기열이 있음을 판정할 수 있다. 반면에, 비-STAT 샘플은 면역검정 분석기 A에 의해 처리될 필요가 있는 것이 아니라, 면역검정 분석기 B 및 화학적 분석기 A에 의해 처리될 것인데, 이들은 처리되기를 기다리는 샘플들의 대기열들을 갖지 않는다. 이러한 시나리오 하에서, 전체적으로 시스템의 제약들의 관점에서 실험실 자동화 시스템의 처리 속도 및 활용도를 최대화하기 위해 STAT 샘플이 처리를 위해 선택되기 전에 비-STAT 샘플이 처리를 위해 선택될 수 있다. 추가적으로, 다운스트림 분석기들에 의한 프로세스를 위해 각각의 샘플을 준비하도록 샘플 준비 시스템에 의해 요구되는 단계들의 수 및 유형이 또한 전체 실험실 시스템을 통한 유동을 최대화하기 위해 샘플의 선택에 고려될 수 있다.

[0014] 본 발명의 실시 형태들에서, PCL 및 접속된 및 비-접속된 분석기들뿐만 아니라 원심분리기(centrifuge)들, 피펫터들 등과 같은 샘플 준비 시스템에 포함된 제어가능한 실험실 장비를 포함하는 실험실 자동화 시스템과 WML 사이에 꾸준한 통신이 있을 수 있다. WML은 접속된 및 비-접속된 분석기들의 상태(예컨대, 분석기들이 가동 중, 비-가동 중, 사용 중, 미사용 중 등임), 또는 기타 제어가능한 실험실 장비의 상태에 대한 업데이트들을 지속적으로 수신할 수 있고, 프로세스 계획들 및 분석기 상태 정보를 PCL에 지속적으로 제공할 수 있다. 이러한 정보를 이용하면, PCL은 환자 샘플들에 대한 스케줄을 그러한 샘플에 대한 루트에 의존하여 생성하도록 독립적으로 동작할 수 있는데, 이는 실험실 자동화 시스템 및 기타 제어가능한 실험실 장비를 통한 샘플의 처리를 최적화할 수 있다.

[0015] 본 발명의 실시 형태들에서, 처리가 시작되는 시간으로부터 샘플이 분석기에 의해 분석된 후의 시간까지의 샘플을 처리하기 위한 시간은 두 개의 시구간들로서 특징지을 수 있다. 제1 시구간(기간 A)은 샘플이 분석을 위해 준비되는 실험실 자동화 시스템의 샘플 준비 시스템을 통해 샘플을 처리하는 속도일 수 있다. 제2 시구간(기간 B)은 준비된 샘플을 처리하여 요구되는 분석기들에서의 테스트 결과를 판정하는 데 필요한 나머지 시간일 수 있다. 본 발명의 실시 형태들은 제1 시구간(기간 A) 및 제2 시구간(기간 B)의 총합을 효과적으로 최소화하는 데 이용될 수 있다.

[0016] 본 발명의 일 실시 형태는, 작업흐름 관리 층을 이용하는 적어도 하나의 프로세서에 의해, 샘플 컨테이너(예컨대, 샘플 튜브) 내의 샘플에 대한 프로세스 또는 루트 계획을 생성하는 단계, 및 루트 계획을 프로세스 제어 층(PCL)에 제공하는 단계를 포함하는 방법에 관한 것이다. 루트 계획은 주어진 순서로 될 수 있는 처리 명령어들의 목록을 포함한다. 방법은 또한, 프로세스 제어 층을 이용하는 적어도 하나의 프로세서에 의해, 샘플 준비 시스템을 통해, 루트 계획에 따라 그리고 샘플 준비 시스템의 이용가능성 및 샘플의 우선순위에 따라, 요구되는 분석기들 각각으로의 최적화된 루트를 판정하는 단계, 프로세스 제어 층(PCL)을 이용하는 단계, 및 최적화된 루트를 이용하여 샘플을 처리하는 단계를 포함한다.

[0017] 본 발명의 다른 실시 형태는 적어도 하나의 프로세서, 및 적어도 하나의 프로세서에 의해 실행가능한 복수의 소프트웨어 컴포넌트들을 저장하는 메모리 디바이스를 포함하는 컴퓨터 장치에 관한 것이다. 복수의 소프트웨어 컴포넌트들은 주어진 순서의 처리 단계들의 목록을 포함하는 프로세스 계획을 생성하도록 동작가능한 작업흐름 관리 층, 및 최적화된 루트를 판정하도록 동작가능한 프로세서 제어 층을 포함한다.

[0018] 본 발명의 다른 실시 형태는 복수의 샘플 처리 서브시스템들, 및 컴퓨터 장치를 포함하는 시스템에 관한 것이다. 컴퓨터 장치는 적어도 하나의 프로세서, 및 적어도 하나의 프로세서에 의해 실행가능한 복수의 소프트웨어 컴포넌트들을 저장하는 메모리 디바이스를 포함한다. 복수의 소프트웨어 컴포넌트들은 프로세스 계획을 생성하도록 동작가능한 작업흐름 관리 층, 및 최적화된 루트를 선택하도록 동작가능한 프로세서 제어 층을 포함한다.

[0019] 본 발명의 다른 실시 형태는, 제1 프로세서와 작업흐름 관리 층을 포함하는 제1 컴퓨터 판독가능 매체를 포함하는 작업흐름 관리 제어기를 포함하는 자동화된 샘플 처리 시스템에 관한 것이다. 시스템은 또한, 작업흐름 관리 제어기에 연결되고 실험실 자동화 시스템의 일부인 LAS 제어기를 포함하는데, LAS 제어기는 제2 프로세서와, 프로세스 제어 층 및 중간 제어 층을 포함하는 제2 컴퓨터 판독가능 매체를 포함한다.

[0020] 본 발명의 다른 실시 형태는, 제1 프로세서와 작업흐름 관리 층을 포함하는 제1 컴퓨터 판독가능 매체를 포함하는 작업흐름 관리 제어기에 의해, 처리될 샘플에 대한 명령어 데이터를 작업흐름 관리 제어기에 연결된 LAS 제어기에 제공하는 단계 - LAS 제어기는 실험실 자동화 시스템의 일부이고, 제2 프로세서와, 프로세스 제어 층 및 중간 제어 층을 포함하는 제2 컴퓨터 판독가능 매체를 포함함 - 를 포함하는 방법에 관한 것이다. 방법은 또한 LAS 제어기에 의해 명령어 데이터를 실행하는 단계를 포함할 수 있다.

[0021] 본 발명의 다른 실시 형태는 프로세스 제어 층을 이용하는 적어도 하나의 프로세서에 의해 루트 레그(route leg)를 중간 제어 층에 제공하는 단계를 포함하는 방법에 관한 것이다. 방법은 또한, 적어도 하나의 프로세서 및 중간 제어 층에 의해, 루트 레그의 서브시스템의 동작 또는 루트 레그의 서브어셈블리 컨테이너의 동작을 제어하도록 하는 명령어들을 생성하는 단계 - 서브어셈블리 컨테이너는 서브어셈블리 컨테이너와 관련된 다수의 서브어셈블리들을 제어함 - 를 포함한다. 방법은 또한 디바이스 제어 층에 의해 명령어들을 실행하는 단계를 포함한다. 일부 실시 형태들에서, 중간 제어 층(MCL)은 추가적으로 루트 레그들을 PCL로부터 수신하여 처리할 수 있고, 이들 루트 레그들은 추가적으로 MCL에 의해 최적화될 수 있다. MCL은 본 발명의 실시 형태들에서 PCL 및 WML에 독립적으로 동작할 수 있다.

[0022] 본 발명의 다른 실시 형태는 적어도 하나의 프로세서 및 적어도 하나의 프로세서에 의해 실행가능한 복수의 소프트웨어 컴포넌트들을 저장하는 메모리 디바이스를 포함하는 컴퓨터 장치를 포함하는 시스템에 관한 것이다. 소프트웨어 컴포넌트는 적어도 하나의 프로세서를 사용하여 루트 레그를 중간 제어 층에 제공하도록 구성되는 프로세스 제어 층, 적어도 하나의 프로세서를 사용하여 루트 레그의 서브시스템의 동작 또는 루트 레그의 서브어셈블리 컨테이너의 동작을 제어하도록 하는 명령어들을 생성하도록 구성된 중간 제어 층을 포함한다. 서브어셈블리 컨테이너는 서브어셈블리 컨테이너와 관련된 다수의 서브어셈블리들을 제어한다. 시스템은 또한 적어도 하나의 프로세서를 사용하여 명령어들을 실행하도록 구성되는 디바이스 제어 층을 포함한다.

[0023] 또 다른 실시 형태에서, WML은 두 개의 개별 제어 층들에 의해 형성된다. 이들 층들은 단일 프로세서 또는 두

개 이상의 상이한 프로세서들을 사용하여 실행될 수 있다. 제1 층은 환자 샘플들에 대해 수행될 테스트들의 유형들에 대한 명령어들을 수신한다. 이어서, 이러한 제1 층은 그러한 테스트들을 수행하는 데 필요한 분석기의 유형을 정의하는 그러한 샘플에 대한 루트 계획을 개발한다. 루트 계획은 오로지 분석기의 유형만을 식별하고, 그러한 테스트들을 수행할 특정 분석기를 식별하지는 않는다. 제2 제어 층은 샘플 준비 시스템 및 개별 분석기들과의 지속적인 통신을 제공한다. 이러한 제2 제어 층은 준비 시스템 및 분석기들에게 이들 유닛들 각각의 상태를 판정하도록 질의한다. 이러한 상태는 각각의 시스템 또는 분석기의 이용가능성, 각각의 시스템 또는 분석기에서의 필요한 시약들 및 공급물들의 양, 및 각각의 시스템 또는 분석기의 보수 및/또는 품질 유형 상태를 포함한다. 이러한 제2 층은 준비 시스템 및 분석기들의 상태에 대한 정보를 샘플 루트 정보와 함께 PCL에 제공하는데, PCL은 이어서 그러한 정보를 이용하여 실험실 시스템을 통해 개별 샘플들에 대한 스케줄을 개발한다. 이러한 실시 형태에 따르면, WML의 제1 층은 미들웨어로 지칭되는 것에 의해 포괄될 수 있으며, 이 경우, 제2 층은 실험실 자동화 시스템들에서 현재 채용되어 있지 않은 추가적인 처리 층이다.

[0024] 본 발명의 이들 및 다른 실시 형태들이 아래에 더욱 상세히 기술된다.

도면의 간단한 설명

[0025] 도 1a는 본 발명의 실시 형태에 따른 하이 레벨 관리 아키텍처의 블록 다이어그램이다.

도 1b는 본 발명의 실시 형태에 따른 관리 아키텍처의 블록 다이어그램이다.

도 1c는 본 발명의 실시 형태에 따른 자동화된 샘플 처리 시스템의 일부 컴포넌트들을 예시하는 블록 다이어그램을 도시한다.

도 2a는 본 발명의 실시 형태에 따른 지향적 비순환 그래프(directed acyclical graph)를 도시한다.

도 2b는 본 발명의 실시 형태에 따른 방법을 예시하는 플로우차트를 도시한다.

도 3은 다양한 레벨 인터페이스들의 다이어그램을 도시한다.

도 4는 컴퓨터 장치의 블록 다이어그램을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0026] 본 발명의 실시 형태들은 임상 실험실에서 환자 샘플들의 작업흐름을 관리하기 위한 개선된 샘플 작업흐름 관리 아키텍처에 관한 것이다. 본 발명의 실시 형태들을 논의하기 전에, 일부 용어들을 논의하는 것이 도움이 될 수 있다.

[0027] 본 명세서에 사용되는 바와 같이, "명령어 데이터"는 샘플 컨테이너 및/또는 샘플 컨테이너 내의 샘플이 처리될 수 있는 방법에 관한 정보를 포함할 수 있다. 일부 실시 형태들에서, 명령어 데이터는 사용자 또는 장치에 의해 실험실 정보 시스템에 제공되는 더 높은 레벨의 명령어들을 포함할 수 있다. 다른 실시 형태들에서, 명령어 데이터는 샘플 또는 샘플 컨테이너를 처리하기 위한 명령어들을 지칭할 수 있는데, 여기서 명령어들은 하나의 소프트웨어 모듈로부터 다른 소프트웨어 모듈로 전달된다.

[0028] "수송 시스템"은 샘플 컨테이너를 수송하는 데 사용될 수 있는 임의의 적합한 하드웨어를 포함할 수 있다. 예시적인 수송 시스템들은 퍽(puck)들, 트랙(track)들, 벨트(belt)들, 그리퍼(gripper)들 등 중 하나 이상을 포함할 수 있다.

[0029] "샘플 컨테이너 데이터"는 샘플 컨테이너 및/또는 컨테이너 내의 샘플을 특징짓는 데이터를 지칭할 수 있다. 샘플 컨테이너 데이터는, 예를 들어 튜브 상의 캡의 색상, 튜브의 크기 또는 형상, 튜브 상의 식별자(예컨대, 튜브 상의 라벨의 바코드) 등을 포함할 수 있다. 샘플 컨테이너 데이터는 샘플(예컨대, 특정 개인으로부터의 혈액, 소변 등)을 식별하는 데이터 및/또는 샘플을 처리하는 방법(예컨대, 일반 처리, 짧은 턴어라운드 등)을 포함할 수 있다. 일부 실시 형태들에서, 샘플 컨테이너가 수송되는 동안에 샘플 컨테이너의 중량 및/또는 샘플 컨테이너 내의 샘플의 액체 레벨이 로봇식 그리퍼에 의해 판정될 수 있다. 이에 관한 추가 상세사항들을 2011년 11월 7일자로 출원된 미국 가특허 출원 제61/556,667호, 2012년 3월 28일자로 출원된 미국 가특허 출원 제61/616,994호, 및 2012년 8월 6일자로 출원된 미국 가특허 출원 제61/680,066호뿐만 아니라, 2012년 11월 7일자로 출원된 PCT/US2012/063931호에서 찾을 수 있는데, 이러한 출원들은 모든 목적들을 위해 전체적으로 본 명세서에 참고로 포함된다.

[0030] "디바이스 커맨드들"은 특정 서브시스템 또는 서브시스템 내의 서브어셈블리들을 제어하도록 디바이스 제어 층

에 의해 실행되는 명령어들을 포함할 수 있다. 그러한 디바이스 커맨드들은 상태(예컨대, 온 또는 오프), 동작 속도 등에 관한 것일 수 있다.

[0031] 샘플 컨테이너들의 그룹의 맥락에서의 "영역"은 유사하게 처리될(예컨대, 모두가 원심분리될) 그리고/또는 유사한 특징부들(예컨대, 모든 혈액 샘플들)을 갖는 컨테이너들의 그룹을 포함할 수 있다.

[0032] "서브시스템"은 명시된 기능을 수행할 수 있는 기기 또는 기기들의 집합체를 포함할 수 있다. 서브시스템들의 예들은 리캡퍼(recapper), 캡퍼(capper), 원심분리기, 및 분취기(aliquoter)를 포함한다.

[0033] "서브어셈블리"는 서브시스템의 일부일 수 있다. 각각의 서브시스템은 하나 이상의 서브어셈블리들을 포함할 수 있다. 예를 들어, "출력" 서브시스템은 그리퍼 및 다수의 드로워(drawer)들을 포함할 수 있다. 그리퍼는 이러한 예에서 서브어셈블리로 간주될 수 있다.

[0034] "서브어셈블리 컨테이너"는 두 개 이상의 서브어셈블리들의 지정된 그룹화를 제어하는 소프트웨어 모듈을 포함할 수 있다. 두 개 이상의 서브어셈블리들은 상이한 서브시스템들의 일부일 수 있다.

[0035] "메모리 디바이스"는 전자 데이터를 저장할 수 있는 임의의 적합한 디바이스일 수 있다. 적합한 메모리 디바이스는 원하는 방법을 구현하도록 프로세서에 의해 실행될 수 있는 명령어들을 저장하는 컴퓨터 판독가능 매체를 포함할 수 있다. 메모리 디바이스들의 예들은 하나 이상의 메모리 칩들, 디스크 드라이브들 등을 포함할 수 있다. 그러한 메모리 디바이스들은 임의의 적합한 전기, 광, 및/또는 자기 동작 모드를 이용하여 동작할 수 있다.

[0036] "프로세서"는 임의의 적합한 데이터 계산 디바이스 또는 디바이스들을 지칭할 수 있다. 프로세서는 함께 작업하여 원하는 기능을 달성하도록 하는 하나 이상의 마이크로프로세서들을 포함할 수 있다.

[0037] "프로세스 계획"은 샘플 상에서 작동될 필요가 있는 프로세스에 관한 정보를 포함할 수 있다. 그것은 원하는 테스트를 수행하는 데 이용가능한 분석기들의 현재 개수 및 그들의 상태에 관한 정보를 포함할 수 있고, 그것은 또한 샘플을 처리하는 데 이용될 수 있는 처리 단계들 및/또는 서브시스템들의 목록을 포함할 수 있다. 예를 들어, 프로세스 계획은 분취기들, 원심분리기들 등과 같이 샘플이 보내질 샘플 준비 시스템 내의 특정 서브시스템들에 관한 정보를 포함할 수 있다. 일부 실시 형태들에서, 프로세스 계획은 샘플을 처리하는 데 사용될 수 있는 정확한 서브시스템들을 명시하지는 않는다. 예를 들어, 실험실 자동화 시스템이 세 개의 원심분리기들 A, B, 및 C를 포함하는 경우에, 프로세스 계획은 샘플을 처리하는 데 이용되는 단계들 중 하나가 원심분리임을 단순히 나타낼 수 있고, 사용할 특정 원심분리기를 명시하지 않을 수 있다. 다른 예에서, 실험실 자동화 시스템이 세 개의 원심분리기들 A, B, 및 C를 포함하는 경우에, 프로세스 계획은 원심분리기 C가 작동불능일 수 있거나 또는 일시적으로 사용되는 중일 수 있거나 또는 달리 이용불가능할 수 있기 때문에 샘플이 오로지 원심분리기 A 또는 B에 의해서만 처리될 수 있음을 단순히 나타낼 수 있다. 그러면, 프로세스 제어 층은 그 테스트 계획을 이용할 수 있고, 원심분리기들 A 및 B 중에서 어떤 것이 분석 프로세스를 통한 샘플의 속도를 최대화하기 위해 샘플을 처리할 최상의 선택안이 되는지 결정할 수 있다.

[0038] "루트"는 샘플이 실험실 자동화 시스템을 통해 취할 수 있는 경로를 지칭할 수 있다. 경로는 특정 세트의 서브시스템들을 통한 특정 순서, 특정 횟수의 처리를 포함할 수 있다. 예를 들어, 실험실 자동화 시스템이 원심분리기들 A 및 B와, 디캡퍼(decapper)들 A 및 B와, 분취기들 A 및 B를 포함하는 경우에, 루트의 예는 샘플을 원심분리기 A, 디캡퍼 B, 및 이어서 분취기 B를 통하여 처리하는 것일 수 있다.

[0039] "레그(leg)"는 루트의 일부를 포함할 수 있다. 레그는 단일 서브시스템 유형 또는 두 개 이상의 서브시스템 유형들 사이의 경로를 포함할 수 있지만, 모든 서브시스템 유형들이 루트에 요구되는 것은 아니다. 예를 들어, 루트가 원심분리기 A, 디캡퍼 B, 및 분취기 B를 포함하는 경우에, 루트의 레그는 디캡퍼 B 및 분취기 B를 포함할 수 있다. 다른 예에서, 루트 레그는 단순히 분취기 B를 포함할 수 있다.

[0040] 본 발명의 실시 형태들에서, 관리 아키텍처는 실험실 정보 시스템(LIS)과 인터페이싱하고, LIS와 임상 실험실 사이의 중재자(mediator)로서 작용한다. LIS 또는 사용자는 샘플들을 처리하는 데 사용되는 테스트 주문들을 관리 아키텍처에 제공한다. 이들 테스트 주문들에 기초하여, 관리 아키텍처는 실험실 직원 또는 실험실 자동화 시스템(LAS) 중 어느 한쪽에 대한 작업흐름 또는 튜브 명령어들(프로세스 계획들로도 지칭됨)을 생성한다. 관리 아키텍처는 또한 질의 시에 테스트 주문들을 실험실 내의 임의의 접속된(또는 "자동화된") 또는 비-접속된(또는 "독립형") 분석기들에 제공한다. 예시적인 LAS들이 2011년 11월 7일자로 출원된 미국 가특허 출원 제61/556,667호 및 2012년 3월 28일자로 출원된 미국 가특허 출원 제61/616,994호에 기술되어 있는데, 이러한 출

원들은 모든 목적들을 위해 전체적으로 본 명세서에 참고로 포함된다.

[0041] LAS 또는 단일 분석기가 샘플을 처리할 때, 샘플 관련 결과들이 관리 아키텍처 내에서 다루어진다. LAS와 관련하여, 결과들은 샘플에 대해 수행되는 주변-분석 처리(peri-analytical processing)에 대응하는 샘플 상태 업데이트들을 포함하는 튜브 데이터를 포함한다. 여기에 사용되는 바와 같이, "주변-분석"은 "사전-분석" 처리 및 "사후-분석" 처리를 포함할 수 있는 처리를 지칭할 수 있다. 단일 분석기와 관련하여, 결과들은 테스트 결과들을 포함할 수 있다. 이들 테스트 결과들의 수신 시에, 관리 아키텍처는 테스트 결과들에 대해 추가 분석들을 수행할 수 있고, 요구된다면, 샘플에 대한 작업흐름을 업데이트할 수 있다. 테스트 결과들을 입증한 후에, 관리 아키텍처는 테스트 결과들을 LIS 또는 사용자에게 포워드한다.

[0042] 도 1a는 본 발명의 실시 형태에 따른 하이 레벨 관리 아키텍처의 블록 다이어그램이다. 도 1a는 작업흐름 관리자 부분(14)과 인터페이싱하는 사용자 인터페이스 부분(12)을 도시한다. 사용자 인터페이스 부분(12)은 실험실 기술자와 같은 사용자와 상호작용하는 부분일 수 있다. 그것은 실험실 정보 시스템(LIS)의 일부일 수 있다. 작업흐름 관리자 부분(14)은 다양한 LAS들에 대한 작업흐름을 관리할 수 있다. 그것은 작업흐름 관리 층을 포함할 수 있는데, 이는 아래에서 더 상세히 기술된다.

[0043] 작업흐름 관리자 부분(14)은 서브시스템 부분(18)과 인터페이싱하는 기기 관리자(16)와 인터페이싱한다. 기기 관리자 부분(16)은 프로세스 제어 층을 포함할 수 있는데, 이는 아래에서 더 상세히 기술된다. 서브시스템 부분(18)은 하나 이상의 서브시스템들(예컨대, 원심분리기, 디캡퍼 등)을 포함할 수 있다. 기기 관리자 부분(16)은 또한 중간 제어 층 및 디바이스 제어 층을 포함할 수 있다.

[0044] 본 발명의 실시 형태들에서, 작업흐름 관리 부분(14)은 가변적/가요적 프로세스 계획을 생성할 수 있는 반면, 기기 관리자 부분(16)은 가변적/가요적 프로세스 계획에 따라 프로세스를 스케줄링한다. 본 발명의 실시 형태들은 다수의 가능한 루트들을 포함하는 가요적 또는 가변적 프로세스 계획의 생성을 허용한다. 프로세스의 스케줄링은 가변적/가요적 프로세스 계획의 생성으로부터 분리될 수 있다.

[0045] 도 1b는 본 발명의 실시 형태에 따른 관리 아키텍처(100)의 블록 다이어그램을 도시한다. 관리 아키텍처(100)는 적어도 세 개의 계층 레벨들인 실험실 레벨(50), 기기 레벨(60), 및 서브시스템 레벨(70)의 데이터를 관리할 수 있는데, 각각의 레벨은 자신의 특정 세트의 데이터를 책임진다. 이들 층들은, 메모리 및/또는 컴퓨터 판독가능 매체 상에 저장되고 하나 이상의 컴퓨터 장치들 상에 존재하는 하나 이상의 프로세서들(예컨대, 데이터 프로세서들)과 함께 작업하는 소프트웨어 컴포넌트들의 형태로 있을 수 있다. 예를 들어, 모든 세 개의 층들(50, 60, 70)이 하나 이상의 프로세서들(예컨대, 마이크로프로세서들)을 갖는 하나의 컴퓨터형 장치 상의 컴퓨터 판독가능 매체 상에 존재할 수 있다. 대안적으로, 세 개의 층들(50, 60, 70)은 세 개의 동작가능한 컴퓨터형 장치들 상에 존재하는 세 개의 컴퓨터 판독가능 매체 상에 존재할 수 있는데, 컴퓨터형 장치들은 각각이 하나 이상의 프로세서들(예컨대, 마이크로프로세서들)을 갖는다. 일부 실시 형태들에서, 작업흐름 관리 층(50(b))은 제1 컴퓨터형 장치(예컨대, 제1 서버 컴퓨터) 내에 존재하는 반면, 프로세스 제어 층(60(a)) 및 중간 제어 층(70(a))은 제2 컴퓨터형 장치(예컨대, 제2 서버 컴퓨터) 내에 존재한다. 제1 및 제2 컴퓨터형 장치들은 서로 통신할 수 있지만, 서로 독립적으로 작동할 수 있다. 제2 컴퓨터형 장치는 LAS(80)의 일부일 수 있다.

[0046] 실험실 레벨(50)에서, 관리 아키텍처(100)는 전체 실험실에 걸쳐서 구성 데이터를 관리하는 것을 책임질 수 있다. 실험실 설정에서, 관리 아키텍처(100)는 전형적으로 하나 이상의 LAS들과 다수의 접속된 및 비-접속된 분석기들(60(b), 60(c))에 의해 요구되는 정보를 관리한다.

[0047] 본 발명의 실시 형태들에서, LAS(80)는 프로세스 제어 층(60(a)), 중간 제어 층(70(a)), 디바이스 제어 층(70(b)), 및 디바이스 펌웨어(70(c))를 포함할 수 있다. 본 발명의 실시 형태들은 또한 많은 LAS들과, 하나 또는 다수의 접속된 및 비-접속된 분석기들(60(b), 60(c))을 포함할 수 있는 한편, 단일 작업흐름 관리 층(50(b))을 이용한다.

[0048] 다수의 자동화 라인들에 걸쳐서 공통적인 실험실 레벨(50) 데이터 항목들이 이러한 레벨에서 구성될 수 있다. 구체적으로, 구성 데이터는 적어도 LAS 입력 영역 정보, 샘플 컨테이너(튜브) 유형 정보, 및 작업흐름 관리 정보를 포함할 수 있다.

[0049] LAS 입력 영역 정보는 LAS 입력 영역 내의 드로워들을 위한 랙(rack)들 및 랙 홀더 플레이트들의 구성에 관한 정보, 및 상이한 랙들 또는 랙 영역들에 대한 명령어들의 맵핑들에 관한 정보를 포함한다. 이들은 실험실 레벨 데이터 항목들로서 정의될 수 있는데, 그 이유는 동일한 LAS 입력 영역 구성이 상이한 LAS들에 대해 이용될 수 있기 때문이다.

[0050] 샘플 컨테이너(튜브) 유형 정보는 실험실에서 이용될 튜브 유형들, 캡 유형들, 및 캡 색상들의 구성을 포함한다. LAS는 이들 튜브 유형들을 처리하기 전에 이들에 대해 트레이닝될 수 있다.

[0051] 작업흐름 관리 정보(예컨대, 명령어 데이터)는 관리 아키텍처(100)가 샘플 튜브들에 대한 프로세스 계획들을 생성하는 데 이용할 수 있는 샘플 처리 규칙들을 포함한다. 다수의 LAS들을 갖는 대규모 임상 실험실들에서, 샘플 처리 규칙들은 LAS마다 다를 수 있다. 이러한 경우에, 관리 아키텍처(100)는 LAS들 각각에 대한 적절한 프로세스 계획들을 생성할 수 있다.

[0052] 실험실 레벨에서 생성되는 구성 데이터는 관리 아키텍처 데이터베이스에 저장될 수 있다. 관리 아키텍처 데이터베이스는 작업흐름 관리 층(50(b)) 또는 임의의 다른 적합한 위치에 존재할 수 있다. 그 결과, 데이터는 전체 실험실 도처에서 이용가능할 수 있다.

[0053] 관리 아키텍처(100)는 또한 그래픽 규칙 엔진(도시되지 않음)을 포함할 수 있는데, 이러한 그래픽 규칙 엔진은 관리 아키텍처(100)의 규칙 편집자로 지칭될 수 있다. 그래픽 규칙 엔진은 작업흐름 관리 층(50(b)) 또는 임의의 다른 적합한 위치에 존재할 수 있다. 규칙 편집자는 사용자에게 실험실 작업흐름을 관리하도록 소정 세트의 규칙들을 한정하는 능력을 제공한다. 규칙들은 두 개의 주요 범주들로 구분될 수 있다: 테스트 결과들을 입증하기 위한 규칙들; 및 샘플들을 처리하기 위한 규칙들.

[0054] 테스트 결과들을 입증하기 위한 규칙들은 테스트 결과들이 관리 아키텍처(100)에 의해 자동-입증될 수 있는 방법을 명시한다. 자동-입증 결과들의 일부로서, 허용가능 결과들을 얻는 데 필요하다면, 샘플을 재작동시키는 것, 샘플 처리 단계들에 추가하는 것, 또는 샘플을 희석시키는 것과 같은 추가 처리가 수행될 수 있도록 하는 규칙들이 제공될 수 있다. 규칙들이 여전히 결정적인 것이 못 되는 경우에, 실험실 조작자가 검사를 위해 샘플을 보유할 수 있도록 하는 규칙이 제공될 수 있다.

[0055] 샘플들을 처리하기 위한 작업 명령어들은 LIS(50(a))로부터 수신된 테스트 주문들에 기초하여 샘플들에 필요할 수 있는 주변-분석 처리를 명시한다. 샘플의 주변-분석 처리는 원심분리, 레벨 감지, 디캡핑, 분취, 및 리캡핑을 포함하지만 이들로 제한되지는 않는다. 주변-분석 처리는 또한 샘플 튜브들을 접속된 분석기들, 출력 트레이들, 및 아카이브(archive)들 또는 저장 설비들로 전송하는 것을 포함한다.

[0056] 기기 레벨(60)에서, 관리 아키텍처(100)는 LAS 하드웨어 구성("사이트 네트워크"로도 지칭됨), 작업흐름 명령어들, 및/또는 프로세스 계획들의 가상 표현, 및 특정 작업 명령어들의 서브어셈블리들(루트 레그들로도 지칭됨)을 포함하는 기기 특정 정보를 관리한다. 또한, 관리 아키텍처(100)는 통계 데이터, 로그 파일들, 및 다양한 서브시스템 구성들을 수집 및 관리한다.

[0057] 사이트 네트워크는 사이트 노드들(버퍼, 처리, 및 수송) 및 노드들을 접속시키는 에지들로 이루어진 그래프일 수 있다. 사이트 네트워크는 LAS의 물리적 레이아웃을 표현할 수 있고, 다양한 하드웨어 구성들을 가상적으로 요약하는 데 이용될 수 있다.

[0058] 서브시스템 레벨(70)에서, 관리 아키텍처(100)는 LAS 내의 개별 서브시스템들 및 서브어셈블리들의 교시로부터 생성되는 데이터를 관리한다. 각각의 서브시스템 및 각각의 서브시스템 내의 대응하는 서브어셈블리들은 수많은 로봇식 정렬 지점들, 임계 값들, 및 구성될 필요가 있는 기타 디바이스 특정 데이터를 가질 수 있다. 예를 들어, 설치 시에, 샘플들이 시스템 상에 놓여 처리되기 전에, 입력 스테이션들, 원심분리기들, 디캡퍼들 등과 같은 개별 자동화 서브시스템들이 사용 전에 교시(예컨대, 서브시스템들에게 전체적인 시스템 내의 그들의 위치를 교시함) 또는 정렬될 수 있다. 교시 단계 동안에 생성된 데이터는 구성 데이터 파일들에 저장될 수 있다. 구성 데이터 파일들은 서브시스템 레벨(70)에서 유지될 수 있는데, 그 이유는 데이터가 오로지 특정 서브시스템들에만 적용되기 때문이다. 관리 아키텍처(100)는 서브시스템 레벨(70) 특정 데이터를 다룰 필요가 없거나 또는 심지어 그를 인식할 필요가 없다. 백업/복원 목적들을 위해, 데이터 파일들은 압축(zip up)될 수 있고, 저장을 위해 관리 아키텍처(100) 내의 적절한 데이터베이스로 전송될 수 있다. 복원 시, 서브시스템 레벨(70)은 관리 아키텍처(100)에게 복원을 행하기 전에 백업 압축된 파일을 다운로드할 것인지 문의할 수 있다.

[0059] 상기 논의로부터 명백한 바와 같이, 관리 아키텍처(100)의 아키텍처 레벨들 내에서, 소프트웨어는 소프트웨어 컴포넌트들의 층들로 추가로 조직화될 수 있다. 레이어드 소프트웨어는 기능이 더 미세한 입도로 추가로 분해될 수 있게 한다. 계층은 레이어드 구조에 내포될 수 있다. 이들 층들은, 메모리 및/또는 컴퓨터 판독가능 매체 상에 저장되고 하나 이상의 컴퓨터 장치들 상에 존재하는 하나 이상의 프로세서들(예컨대, 데이터 프로세서들)과 함께 작업하는 소프트웨어 컴포넌트들의 형태로 있을 수 있다.

[0060] 본 발명의 일부 실시 형태들에서, 관리 아키텍처(100)는 네 개의 제어 층들을 포함하는데, 각각의 레벨이 임상 실험실에서 특정 프로세스들을 제어하는 것을 책임진다. 이들은 작업흐름 관리 층(WML)(50(b)), 프로세스 제어 층(PCL)(60(a)), 중간 제어 층(MCL)(70(a)), 및 디바이스 제어 층(70(b))을 포함한다.

[0061] WML(50(b))은 상측 레벨일 수 있거나 또는 사용자에게 관찰될 수 있어서, 그의 규칙 엔진에 명시된 샘플 처리 규칙들을 활용하여 샘플 상에서 작동될 프로세스 계획을 결정하게 할 수 있다. 다수의 LAS 라인들을 갖는 실험실들의 경우에, WML(50(b))은 각각의 라인이 상이한 작업흐름 관리 방식으로 구성되게 할 수 있고, 다수의 활성 작업흐름들을 동시에 관리할 수 있다. 추가로, WML(50(b))은 LIS 인터페이스(50(a)), 품질 제어(QC), 결과 관리, 샘플 추적, 백업 및 복원, 기기 상태, 테스트 순서정렬, 구성, 온라인 도움말, 사용자 인증, 및 기기 발견(접속된 서브시스템들을 식별하기 위해 시스템의 초기화 단계에서 작동되는 초기 검색)을 포함한 중요 소프트웨어 특징들을 다룰 수 있다. 도시된 바와 같이, LIS 인터페이스(50(a))는 WML(50(b))에 테스트 주문들을 제공할 수 있고, WML(50(b))로부터 테스트 결과들을 수신할 수 있다.

[0062] WML(50(b))은 샘플들(구성 데이터)을 처리하도록 모든 LAS 및 접속된 디바이스들에 의해 이용되는 실험실 레벨(50) 정보를 저장할 수 있다. 시동 시, WML(50(b))은 구성 데이터를 PCL(60(a))로 전달할 수 있고, PCL(50(a))에게 시동 후의 구성 데이터에 대한 임의의 추가적인 변화들을 통지할 수 있다.

[0063] 동작 동안, WML(50(b))은 단일 샘플 컨테이너에 대한 프로세스 계획들을 생성하고, 이러한 프로세스 계획을 PCL(60(a))에 제공한다. WML(50(b))은 그의 기능을 가능하게 하는 여러 상이한 소프트웨어 컴포넌트들을 포함할 수 있다.

[0064] 도 1b에 도시된 바와 같이, WML(50b)은 접속된 분석기들(60(b)) 및 비-접속된 분석기들(60(c))과 통신 상태로 될 수 있다. 테스트 주문들이 분석기들(60(b), 60(c))에 제공(예컨대, 송신)될 수 있고, 테스트 결과들 및 다른 분석기 상태 정보가 WML(50(b))에 역으로 제공될 수 있다. 다양한 분석기들(60(b), 60(c))의 상태에 관한 정보가 WML(50(b))로부터 PCL(60(a))에 제공될 수 있고, 이러한 정보는 최적의 루트(예컨대, 샘플의 우선순위, 분석기들의 이용가능성, 및 시스템 내의 서브어셈블리들 및 서브컴포넌트들의 상태를 취하는 루트)를 생성하여 실험실 자동화 시스템(80)의 처리 용량 및 속도를 최대화하도록 PCL(60(a))에 의해 활용될 수 있다.

[0065] 본 발명의 실시 형태들은 작업흐름 관리 제어기 및/또는 LAS 제어기를 사용하여 본 명세서에 기술된 기능들을 달성할 수 있다. 작업흐름 관리 제어기는 실험실 내의 모든 접속된 및 비-접속된 분석기들에 대한 데이터 및 작업흐름 관리를 다루도록 설계된 서버 컴퓨터일 수 있다. 작업흐름 관리 제어기는 모델을 유지하고, 데이터베이스를 사용하여 그를 지속시킨다. 그것은 실험실 정보 시스템(50(b)) 및 작업흐름 관리 층(50(b))을 작동시킬 수 있다.

[0066] LAS 제어기(도시되지 않음)는 작업흐름 관리 제어기가 고장난 경우에 기기가 독립형 컴퓨터로서 작동할 수 있게 한다. 이에 따라, 그것은 작업흐름 관리 제어기 컴퓨터와 구별되는 서버 컴퓨터일 수 있고, 그것은 프로세스 제어 층(60(a)) 및 중간 제어 층(70(a))을 포함할 수 있다. 디바이스 관리자는 LIS(50(a)), 레거시 분석기들, 및 기타 시스템들과 같은 외부 시스템들과의 통신을 다룰 수 있다. 재검토 클라이언트(예컨대, 작업흐름 관리 층(50(b)) 내에 있음)는 정보를 사용자에게 제공할 수 있고, 사용자가 입력을 모델에 제공할 수 있게 한다. 재검토 클라이언트는 모델과 함께 작업하지만, 모델에 대한 모든 변화들은 작업흐름 관리 제어기로의 비동기식 요청들에 의해 수행될 수 있다.

[0067] 본 발명의 실시 형태들은, 제1 프로세서와 작업흐름 관리 층을 포함하는 제1 컴퓨터 판독가능 매체를 포함하는 작업흐름 관리 제어기, 및 작업흐름 관리 제어기에 연결된 하나 이상의 LAS 제어기들을 포함하는 자동화된 샘플 처리 시스템을 포함할 수 있다. LAS 제어기들 각각은 다른 프로세서와, 프로세스 제어 층 및 중간 제어 층을 포함하는 다른 컴퓨터 판독가능 매체를 포함한다.

[0068] 도 1c는 본 발명의 실시 형태에 따른 자동화된 샘플 처리 시스템의 일부 하드웨어 컴포넌트들을 예시하는 블록 다이어그램을 도시한다. 시스템은 작업흐름 관리 층(50(b))을 동작시키는 작업흐름 관리 제어기(806)를 포함한다. 작업흐름 관리 제어기(806)는 클라이언트 단말기(804)에 연결될 수 있는데, 이러한 클라이언트 단말기는 사용자가 데이터를 시스템 내에 입력하는 데 그리고 데이터를 시스템으로부터 얻는 데 사용될 수 있다.

[0069] 작업흐름 관리 제어기(806)는 제1 프로세스 제어 층(60(a)-1) 및 제1 중간 제어 층(70(a)-1)을 동작시키는 제1 LAS 제어기(808(a))에 연결될 수 있다. 작업흐름 관리 제어기(806)는 또한 제2 프로세스 제어 층(60(a)-2) 및 제2 중간 제어 층(70(a)-2)을 동작시키는 제2 LAS 제어기(808(b))에 연결될 수 있다. 제1 및 제2 LAS 제어기들(808(a), 808(b))은 제1 및 제2 프로세서들과, 이들 프로세스들과 관련된 제1 및 제2 컴퓨터 판독가능 매체들

(전술된 소프트웨어 층들을 저장하기 위함)을 포함할 수 있다.

[0070] 제1 LAS 제어기(808(a))는 적어도 제1 서브시스템(820(a)) 및 제2 서브시스템(820(b))을 포함하는 제1 복수의 서브시스템들과 통신 상태로 될 수 있다. 제2 LAS 제어기(808(b))는 적어도 제3 서브시스템(820(c)) 및 제4 서브시스템(820(d))을 포함하는 제1 복수의 서브시스템들과 통신 상태로 될 수 있다.

[0071] 두 개의 LAS 제어기들이 도시되어 있지만, 본 발명의 실시 형태들은 두 개 초과의 LAS 제어기들을 포함할 수 있다. 또한, 두 개의 서브시스템들이 각각의 LAS 제어기에 대해 도시되어 있지만, 두 개 초과의 서브시스템들이 각각의 LAS 제어기와 관련될 수 있다.

[0072] 도시된 바와 같이, 도 1b의 하드웨어 구성은 다수의 이점들을 갖는다. 예를 들어, 작업흐름 관리 제어기(806)가 작동불능이 되는 경우, 이는 제1 및/또는 제2 LAS 제어기들(808(a), 808(b))의 동작에 영향을 미치지 않을 것이다. LAS 제어기들은 작업 명령어들을 서브시스템들에 여전히 제공하여, 작업흐름 관리 제어기(806)가 작동불능인 경우에도 샘플들을 처리할 수 있다.

[0073] 도 1b를 다시 참조하면, 관리 아키텍처(100)의 WML(50(b))은 그의 규칙 엔진에 정의된 소정 세트의 샘플 처리 규칙들에 기초하여 샘플 튜브들에 대한 프로세스 계획들을 생성하는 것을 책임질 수 있다. WML(50(b))은 PCL(60(a))에 의한 샘플 튜브 도달의 통지를 수신할 시에 특정 샘플 튜브에 대한 이러한 프로세스 계획을 PCL(60(a))에게 제출한다. 추가로, 기존 프로세스 계획들의 재생성은 튜브 상태들 또는 시스템 상태들 중 어느 한쪽의 변화들이 원래의 프로세스 계획에 영향을 미치는 경우에 바람직할 수 있다. 일부 실시 형태들에서, 프로세스 계획은 노드들 및 루트들을 포함하는 그래프로서 묘사될 수 있는데, 여기서 각각의 노드는 PCL에게 샘플에 대해 원하는 처리를 통지하는 명령어를 포함한다. 명령어는 서브시스템에서 샘플에 대해 수행되는 작용을 표현할 수 있다. 예를 들어, 명령어 "원심분리"는 샘플이 원심분리 서브시스템에서 원심분리됨을 나타낸다. 하기의 목록은 LAS에 의해 지원되는 예시적인 명령어들을 포함한다: 원심분리, 디캡핑, 흡인, 레이블, 처분, 리캡핑, 분류 또는 저장 등.

[0074] 전형적으로, 프로세스 계획은 샘플에 대해 수행될 필요가 있는 프로세스들의 목록을 포함한다. 프로세스 계획에서의 하나 이상의 프로세스들을 수행할 수 있는 다수의 서브시스템들이 존재하는 경우에는, 샘플이 프로세스 계획을 이행하도록 취할 수 있는 다수의 잠재적인 루트들이 존재한다. 이는 도 2a에 도시된 것과 같은 지향적 비순환 그래프로서 그래픽으로 표현될 수 있다. 다른 유형들의 표현들이 본 발명의 다른 실시 형태들에서 사용될 수 있다. 도 2a의 비순환 그래프에 의해 예시된 바와 같이, 본 발명의 실시 형태에 따른 프로세스 계획은 가요적이고 가변적인데, 이는 상이한 샘플들이 서브시스템들의 상이한 조합들에 의해 처리될 수 있기 때문이다. 선택된 세트의 서브시스템들은 다수의 인자들에 기초하여 판정될 수 있다.

[0075] 도 2a에 도시된 바와 같이, 프로세스 계획(200)은 샘플 튜브에 대한 시작 지점(202)을 가질 수 있고, 이어서 샘플 튜브가 원심분리(204)될 수 있다. 원심분리 후에, 튜브는 디캡퍼(206)로 이동할 수 있다. 디캡퍼(206) 후에, 프로세스 계획(200)은 샘플 튜브가 제1 면역검정 시스템(208) 또는 제2 면역검정 시스템(210) 중 어느 하나로 이동할 수 있음을 나타낼 수 있다. 예시적인 면역검정 시스템들은 베크만 컬터, 인크.(Beckman Coulter, Inc.)의 DxI 면역검정 시스템들을 포함할 수 있다. 면역검정 시스템은 샘플 준비 스테이션, 분취 스테이션 등을 포함할 수 있다. 제1 및 제2 면역검정 시스템들(208, 210) 중 하나에 의해 처리된 후에, 이어서 샘플은 제1 분석기(212) 및 제2 분석기(214) 중 하나로 이동할 수 있다. 예시적인 분석기들은 베크만 컬터, 인크.의 AU 680 계열의 분석기들을 포함할 수 있다. 제1 분석기(212) 또는 제2 분석기(214) 중 어느 하나에 의해 처리된 후에, 샘플 튜브는 리캡퍼(216)로, 그리고 이어서 일시 저장 유닛(218)으로 이동할 수 있다. 물론, 추가적인 처리 노드들이 본 발명의 실시 형태들에 따른 프로세스 계획들에 존재할 수 있다. 이러한 특정 예에서는, 샘플이 처리되도록 취할 수 있는 네 개의 잠재적인 루트들이 존재한다. PCL(60(a))에 의해 선택된 특정 루트는 실험실 자동화 시스템(80)을 통해서 가장 빠른 처리를 가져오는 루트일 수 있다. 이와 관련하여, 실험실 자동화 시스템(80) 내의 PCL(60(a))은 WML(50(b))에 독립적으로 동작할 수 있는데, 이는 그것이 실험실 자동화 시스템(80)을 통해 최적의 루트를 판정하기 때문이다.

[0076] 도 1a 및 도 2a를 다시 참조하면, PCL 스케줄러(60(a))는, 프로세스 계획을 분석하는 것, 그리고 이어서 샘플에 대한 가장 최적인 루트를 선택하여 턴어라운드 시간(Turn-Around-Time: TAT) 또는 처리량 요건들 중 어느 한 쪽을 충족시키는 것을 책임질 수 있다. 샘플 튜브가 처리를 위해 프로세스 계획에서의 각각의 서브시스템에서 정지할 때, 새로운 샘플 튜브 정보가 생성될 수 있다. 예를 들어, 원심분리(204)에서 처리한 후에, MCL(70(a))은 샘플 튜브 정보에서의 스핀 상태를 업데이트할 수 있고, 업데이트된 정보를 PCL(60(a))로 포워드할 수 있는 반면, PCL(60(a))은 지속성을 위해 튜브 정보를 WML(50(b))로 포워드할 것이다. 이러한 피드백(예컨대, 튜브 정

보)을 이용하면, WML(50(b))은 프로세스 계획들을 계속해서 업데이트 및 최적화할 수 있는 반면, PCL(60(c))은 샘플에 대한 최적의 루트 계획일 수 있는 것에 대한 그의 결정들을 계속해서 업데이트할 수 있다. 종래의 시스템들(정적인 방식으로 동작할 수 있음)과는 달리, 본 발명의 실시 형태들에 따른 시스템 아키텍처는 실험실 자동화 시스템의 처리 속도 및 활용도를 최대화하는 방식으로 동적으로 동작한다.

[0077] 도 1b에 도시된 바와 같이, WML(50(b))은 프로세스 계획을 포함하는 튜브 명령어들을 PCL(60(a))에 제공할 수 있고, 테스트 결과들을 PCL(60(a))로부터 수신할 수 있다. PCL(60(a))은 추가적으로 튜브 루트 정보를 MCL(70(a))에 제공할 수 있고 튜브 데이터를 그로부터 수신할 수 있다.

[0078] MCL(70(a))은 서브시스템 상태의 변화들(즉, 온라인, 오프라인, 이용가능, 이용불가능)을 PCL(60(a))에게 통지할 수 있는데, PCL(60(a))은 이러한 정보를 이용하여, 영향받은 서브시스템들과 관련된 명령어들의 이용가능성을 WML(50(b))에게 통지한다. LAS 상의 단 하나의 서브시스템에 대해, 서브시스템이 이용불가능해진 경우에, PCL(60(a))은 그 서브시스템과 관련된 명령어가 또한 이용불가능함을 WML(50(b))에게 통지할 수 있다. LAS 상의 동일한 서브시스템들 중 하나 초과의 것이 이용가능한 경우에, 하나의 서브시스템의 손실은 제2 서브시스템의 이용가능성으로 인해 명령어 이용가능성에 영향을 미치지 않을 수 있다. 명령어 이용가능성에 기초하여, WML(50(b))은 프로세스 계획들을 생성 또는 수정하는 데 적절한 정보를 가질 것이다.

[0079] 전체적으로, WML(50(b))은 그 아래의 다른 제어 층들의 전체 수명에 걸쳐서 지속될 수 있다. 샘플 작업흐름 관리, 프로세스 계획들, 튜브 정보 등이 WML(50(b))과 PCL(60(a)) 사이에서 전송 및 수신될 수 있다(이들은 이어서, 적절하다면, 관련 데이터를 MCL(70(a)) 등에 중계할 것이다). WML(50(b))은 그것이 MCL(70(a))에게 지시하는 데 이용하는 스케줄, 루트, 및 루트 레그를 생성할 것을 PCL(60(a))에게 지시하는 프로세스 계획을 생성할 수 있다. 이어서, MCL(70(a))은 적절한 하드웨어 컴포넌트들을 이동시켜 튜브(들)의 전달을 실행할 것을 DCL(70(b))에 명령한다.

[0080] 일부 실시 형태들에서, MCL(70(a))은 그것이 수신하는 루트 레그를 추가로 최적화할 수 있다. 예를 들어, 루트 레그는 다수의 서브어셈블리들을 추가로 포함할 수 있는 서브시스템을 포함할 수 있고, 이들 서브어셈블리들의 동작 또는 선택은 MCL(70(a))에 의해 최적화될 수 있다. 예시적으로, 루트 레그는 명령어 "원심분리"를 포함할 수 있다. 원심분리는 원심분리기, 원심분리 로봇, 어댑터 그리퍼, 및 셔틀을 포함하는 서브어셈블리들을 포함할 수 있다. MCL(70(a))은 특정 원심분리 로봇, 셔틀, 또는 어댑터 그리퍼를 추가로 선택하여 가장 빠른 가능한 처리가 달성되게 할 수 있다. MCL(70(a))은 또한 특정 명령어들을 DCL(70(b))(명령어들을 특정 디바이스 펌웨어(70(c))에 제공할 수 있음)에 제공하여 특정 서브어셈블리들의 동작을 제어할 수 있다.

[0081] 이러한 처리 체인 전반에 걸쳐서, 하나의 층이 층 아래로 처리를 인계함에 따라, 그것은 하위 층들에서 발생하고 있는 처리와 동시에 다른 작용들을 처리하도록 진행할 수 있다.

[0082] PCL(60(a)) 스케줄러는 사이트 네트워크 및 프로세스 계획들을 이용하여 가장 최적인 루트를 판정하고, 이어서 그에 따라 시스템 목적들, 예컨대 루트 스케줄을 생성할 때 샘플 처리량을 최대화하면서 TAT들을 최소화하는 것을 충족시키도록 그 루트를 스케줄링한다. 일부 경우들에 있어서, 다수의 처리 사이트들이 WML 명령어를 충족시키는 데 이용가능한 경우에, PCL(60(a))은 이들 사이트들의 활용도를 최적화하기 위해 부하 밸런싱(balancing)을 수행할 수 있다.

[0083] 또한, PCL(60(a))은 WML(50(b))로부터 제출된 하이 레벨 프로세스 계획들을 중단하는 것, 그리고 MCL(70(a))에 의해 제어되는 디바이스에 대한 더 상세한 루트 레그들을 생성하는 것을 책임질 수 있다. 그렇게 하기 위해, PCL(60(a))은 이들 처리 사이트들의 상태(예컨대, 수송 시간, 대기열, 폐기물 처분 레벨, 소모재 레벨 등)를 MCL(70(a))에 대한 하나 이상의 루트 레그들을 도출하는 데 포함시킨다. PCL(60(a))은 사이트 노드들을 갖는 WML(50(b)) 프로세스 계획들을 사이트 네트워크 그래프로부터 증대시킴으로써, 예를 들어 처리 사이트들을 수송 및 버퍼 사이트들로 표현함으로써 루트 레그들을 구축한다. 최종 루트 레그는 샘플 튜브가 가로지를 필요가 있는 사이트 노드들의 목록을 포함하는데, 여기서 각각의 사이트 노드는 샘플 튜브 상에서 수행될 작용을 표현한다.

[0084] 중간 제어 층(MCL)(70(a))은 서브시스템 내의 LAS 서브시스템들 및 서브어셈블리들의 상태를 제어한다. 하드웨어 어셈블리는 더 복잡한 서브시스템들을 구축하는 데 재사용될 수 있다. 예를 들어, "출력"으로 지칭되는 서브시스템은 튜브 갠트리 로봇들 및 드로워들과 같은 서브어셈블리들을 포함할 수 있다.

[0085] 일부 실시 형태들에서, MCL(70(a))은 "서브어셈블리 컨테이너들"의 개념을 이용하여 공통 리소스들을 공유하는 서브어셈블리들을 그룹화할 수 있다. 본 발명의 일부 실시 형태들은 중간 제어 층에 의해 명령어들을 서브어셈

블리 컨테이너에 제공하는 단계를 포함하는 방법에 관한 것이다. 서브어셈블리 컨테이너는 서브어셈블리 컨테이너와 관련된 다수의 서브어셈블리들을 제어한다. 방법은 또한 서브어셈블리 컨테이너에 의해 명령어들을 실행하는 단계를 포함한다. 다수의 서브어셈블리들은 두 개 이상의 상이한 서브시스템들의 일부들을 포함한다.

[0086] 예시적으로, 공통 로봇 팔을 공유하는 두 개의 서브어셈블리들이 컨테이너 내로 배치될 수 있어서, 컨테이너는 두 개의 서브어셈블리들 사이의 로봇 팔 이동을 조정할 수 있다. 컨테이너가 어떤 의미에서는 소정 그룹의 서브어셈블리들을 둘러싸는 "랩퍼(wrapper)"이기 때문에, 메시징이 특정 서브시스템을 위해 예정된 경우라 하더라도 컨테이너와의 물리적 통신이 존재한다. 서브어셈블리 컨테이너는 모든 사용되는 서브어셈블리 제어기들을 호스팅하며, 서브어셈블리 컨테이너가 PCL(60(a)) 요청들을 이행할 수 있다는 의미에서는 자율적이다. 그러한 것은 서브어셈블리 컨테이너가 튜브를 하나의 사이트 노드로부터 다른 사이트 노드로 이동시킬 수 있고 그것이 루트 레그에서 요청된 명령어들에 따른 프로세스 단계들을 실행할 수 있다는 것을 의미한다. 서브어셈블리 컨테이너는, 모든 서브어셈블리들의 임의의 하드웨어 움직임 또는 초기화에 대한 충돌 회피와 같이, 호스팅된 서브어셈블리 제어기들을 조정하는 추가적인 서비스들을 제공하여 서브시스템의 일관되고 에러가 없는 상태를 보장하게 할 수 있다.

[0087] 일부 실시 형태들에서, 특히 복잡한 구조들을 갖는 서브어셈블리 컨테이너들(그러한 컨테이너에 랩핑된 다수의 서브어셈블리들)에 대해, PCL(60(a))에 의해 MCL(70(a))에 제공된 루트 레그는 PCL(60(a))에 의해 모든 상세사항들이 스케줄링되지는 않는다. MCL(70(a))은 서브어셈블리 컨테이너에 대한 루트 레그들을 최적화하는 스케줄 능력들을 추가적으로 갖는다. 그것은, 서브어셈블리 컨테이너에 관련된 사이트 네트워크의 부분 지식을 가지며, 프로세스 계획을 위해 PCL(60(a))이 행하는 것과 유사한 방식으로 루트 레그를 최적화한다. 스케줄러 내의 이러한 계층 구조는 시스템이 단일 단계에서 루트들을 최적화하는 단일 스케줄링 인스턴스를 사용하는 시스템보다 더 빠르고 더 정밀한 방식으로 처리를 스케줄링하게 한다.

[0088] MCL(70(a))은 루트 레그들을 PCL(60(a))로부터 수신하고, 루트 레그들을 그것이 DCL(70(b))로 전송하는 디바이스 커맨드들에 맵핑하며, 그것은 이러한 루트 레그에 기초하여 PCL(60(a))과의 상호작용 없이 자율적으로 튜브를 처리한다. 특히, 루트 스케줄은 PCL(60(a))에서 여러 (중첩하는) 루트 레그들로 분할되어 다양한 서브어셈블리 컨테이너들을 수용한다(따라서, 루트 레그는 루트 스케줄의 일부일 수 있다). 루트 레그는 이러한 서브어셈블리 컨테이너에 대한 모든 필수 노드들을 포함한다. PCL은 루트 레그를 적절한 서브어셈블리 컨테이너로 전송한다. 루트 레그는 튜브 또는 다른 유형의 컨테이너가 서브어셈블리 컨테이너에 물리적으로 도달하기 전에 이용가능하여, 통신 시간 오버헤드를 회피하게 할 수 있다.

[0089] MCL(70(a))은, 예를 들어 그것이 사이트 네트워크 그래프를 가로지름에 따른 튜브의 진행 또는 에러를 나타내는 샘플 상태를 PCL(60(a))에 제공한다. 서브어셈블리 컨테이너를 떠나는 튜브는 적절한 MCL(70(a)) 서브시스템들의 범주를 벗어날 수 있다.

[0090] 관리 아키텍처(100)는 하나의 PCL(60(a))이 다수의 MCL들(70(a))과 통신하게 하는 유연성을 제공한다. 이러한 일-대-다 관계는 실험실에서 다양한 하드웨어 구성들을 매칭하도록 크기를 조정할 수 있다. 튜브 정보는 그것이 튜브를 처리할 필요가 있을 때 MCL(70(a))로 이송될 수 있으며, 처리가 완료되거나 튜브에 관한 임의의 데이터가 변화했을 때 PCL(60(a))로 다시 이송된다.

[0091] 선단(tip)들, 캡들 등의 폐기물 및 공급물 상태를 포함한 하드웨어 서브시스템의 상태가 PCL(60(a))로 이송될 수 있다. 포함될 수 있는 기타 서브시스템 상태들은 다음의 서브시스템의 상태를 포함한다; 작동, 초기화, 중지, 또는 에러. 또한, 처리 제어 메시지들이 서브시스템들을 조정하도록 이송되어서 TAT 및 시스템 처리량 활용 요건들을 보장하게 할 수 있다.

[0092] DCL(70(b))은 LAS 내의 하드웨어 컴포넌트들을 제어하는 하드웨어 제어 층일 수 있다. DCL(70(b))은 통신 인터페이스를 MCL(70(a))에 제공하여, 상이한 디바이스들에 의해 이용되는 전용 프로토콜들 및 상이한 인터페이스들을 랩핑한다. 그것은, 특정 인터페이스 유닛들을 추가함으로써 확장될 수 있고 임의의 프로토콜 및 작업흐름을 IEC 61131-3 순응 프로그래밍 언어로 구현할 수 있는 모듈러 시스템으로서 설계될 수 있다.

[0093] DCL(70(b))은 로우 레벨 커맨드들을 MCL(70(a))로부터 수신할 수 있고, 상이한 디바이스들에 의해 이용되는 전용 프로토콜들 및 상이한 인터페이스들을 이용하여 하드웨어의 직접 제어를 제공한다. DCL(70(b))은 커맨드들을 실행하고, 하드웨어를 제어하며, 커맨드 결과들(예컨대, 액추에이터 위치들, 센서 데이터 등)로 MCL(70(a))에게 응답한다. DCL(70(b))은 또한 커맨드들을 제공하고, 커맨드 결과들을 디바이스 펌웨어(70(c))로부터 수신한다.

[0094] 도 1b에 도시된 바와 같이, 디바이스 펌웨어(70(c))는 핸드셰이크 커맨드들 및 다른 유형들의 커맨드들 또는 메시지들을 교환함으로써, 접속된 분석기들(60(b))과 통신할 수 있다.

[0095] 일부 실시 형태들에서, LAS 컴포넌트들은 서로 비동기식으로 제어될 수 있다. 그들은 DCL(70(b))에 의해 구체적으로 명령되지 않는다면 유휴 상태일 수 있다. 이는 상위 레벨들로부터 전송된 커맨드의 결과일 수 있다. 하드웨어 컴포넌트들과의 동기화는 DCL(70(b))에서 다루어질 수 있어서 충돌들(예컨대, 로봇 움직임들이 서로 중첩할 수 있는 경우에 로봇 움직임들이 동기화될 필요가 있음) 또는 기타 기계적 장애들을 방지할 수 있다. 그렇지 않다면, 하드웨어는 서로 동시적으로 모두 제어될 수 있는데, 다시 말해, 하나의 하드웨어 컴포넌트는 그것이 그의 작용을 시작할 수 있기 전에 어떠한 다른 것도 기다릴 필요가 없다.

[0096] 일 실시 형태에서, DCL(70(b))은 두 개의 일반(generic) DCL-분석기 인터페이스들을 제공하는데, 하나는 직접 이송 샘플링(Direct Transport Sampling: DTS) 분석기들을 위한 것이고, 하나는 랙 기반 분석기(rack based analyzer: RBU)들을 위한 것이다. 이들 일반 인터페이스들은 분석기 하드웨어 변화들에 맞서서 MCL(70(a)), PCL(60(a)), 및 WML(50(b))을 보호한다. DTS 서브어셈블리 컨테이너는 캐리어의 고정(locking) 및 해제(releasing)를 책임질 수 있다. 일반 DCL-DTS 분석기 인터페이스는 모든 가능한 DTS 분석기들로 튜브 핸드셰이크를 다룰 수 있다. DTS 서브어셈블리 컨테이너가 캐리어를 고정한 후에, DCL(70(b))은 흡인할 것을 분석기에게 알릴 수 있다. 그 후에, 분석기는 흡인 프로세스가 완료되었음을 DCL(70(b))에게 알릴 수 있다. 이어서, DTS 서브어셈블리 컨테이너가 캐리어를 해제할 수 있다. RBU 서브어셈블리 컨테이너는 튜브들을 수송 시스템으로부터 랙으로 그리고 그 역으로 수송하는 것을 책임진다. RBU-분석기 인터페이스는 랙 핸드셰이크를 다룰 수 있다. DCL(70(b))은 랙이 분석기에 반입할 준비가 되는 때를 분석기에게 알린다. 분석기는 랙이 분석기로부터 RBU로의 수송을 위한 준비가 되는 때를 DCL(70(b))에게 알린다.

[0097] DCL(70(b))(사이트 그래프의 추상적 노드들과 작업하지 않음)은 제외하고, 모든 소프트웨어 레벨들(WML, PCL, MCL)은 사이트 네트워크 그래프의 상이한 뷰를 상이한 레벨들의 입도로 가질 수 있다.

[0098] 본 발명의 실시 형태들에서, WML(즉, 루트 계획자)이 사이트 그래프의 거시적 뷰를 가질 수 있다. 예를 들어, 일부 경우에, 사이트 그래프의 "주요" 노드들/사이트들만이 작업흐름 관리 층에 가시적이다. WML은 이들 "주요" 노드들 사이의 접속부들(에지들)을 알고 있고, 샘플들이 이들 노드들 사이에서 이송될 수 있는 방향을 알고 있다. 그러한 이송에는 시간에 관한 어떠한 정보도 필요하지 않다.

[0099] 본 발명의 실시 형태들에서, PCL(즉, 스케줄러)이 사이트 네트워크 그래프의 더 상세한 뷰를 가질 수 있다. 그것은 WML보다 더 많은 노드들을 관찰할 수 있다. 예를 들어, 이송 노드들은 PCL에 가시적일 수 있지만, 일부 실시 형태들에서는 WML에 가시적이지 않을 수 있다. 이들 노드들 사이의 에지들뿐만 아니라 허용된 방향들이 또한 PCL에 알려져 있을 수 있다. PCL은 또한 이들 노드들 사이에서 샘플들을 이동시키는 데 필요한 시간을 알고 있다.

[0100] 본 발명의 실시 형태들에서, MCL은 서브시스템(예컨대, 디캡퍼 모듈 또는 원심분리 모듈)의 활동을 제어할 수 있다. 각각의 서브시스템 노드들만이 MCL에 가시적이지만, MCL은 PCL보다 그러한 서브시스템 노드들의 심지어 더 상세한 뷰를 가질 수 있다. 예를 들어, PCL 또는 WML에 가시적이지 않게 될, MCL에 대한 추가적으로 가시적인 노드는 서브시스템 내의 이송 레인의 우회기(diverter)일 수 있다.

[0101] **예시적인 프로세스 흐름**

[0102] 본 발명의 일부 실시 형태들은 방법들에 관한 것이다. 일 실시 형태에서, 방법은 샘플 컨테이너 내의 샘플에 관한 명령어 데이터를 수신하는 단계, 작업흐름 관리 층을 이용하는 적어도 하나의 프로세서에 의해, 샘플에 대한 프로세스 계획을 생성하는 단계, 및 프로세스 계획을 프로세스 제어 층에 제공하는 단계를 포함한다. 프로세스 제어 층 및 적어도 하나의 프로세서는 프로세스 계획을 이용하여 최적화된 루트를 판정한다. 최적화된 루트는 중간 제어 층에 제공되는데, 여기서 적어도 하나의 프로세서 및 중간 제어 층은 수송 시스템이 샘플 컨테이너를 선택된 루트를 따라 수송하게 하도록 동작가능하다.

[0103] 도 1a의 관리 아키텍처를 이용한 데이터 흐름의 시퀀스가 임상 작업흐름과 함께 기술될 수 있다. 작업흐름의 단계들이 도 2b에 도시되어 있다. 본 발명의 실시 형태들은 도 2b에 기술된 처리 단계들로 제한되지 않고, 하나 이상의 단계들이 본 발명의 실시 형태들에서 생략 또는 추가될 수 있다.

[0104] 단계 701에서, 의료 또는 실험실 직원이 실험실 정보 시스템(50(a))에서 또는 작업흐름 관리 층(50(b))에서 샘플들에 대한 테스트 주문들을 입력한다. 테스트 주문들은 처리될 샘플(예컨대, 혈액) 및/또는 튜브(예컨대, 테

스트 튜브)의 유형, 원하는 분석의 유형, 및/또는 샘플의 우선순위(예컨대, STAT 또는 짧은 턴어라운드 시간)를 나타낼 수 있다.

[0105] 단계 702에서, 실험실 정보 시스템(50(a))은 테스트 주문들을 작업흐름 관리 층(50(b))에 다운로드한다. 본 발명의 다른 실시 형태들에서, 작업흐름 관리 층(50(b))은 테스트 주문들에 대한 질의를 실험실 정보 시스템(50(b))으로 전송할 수 있다(단계 703).

[0106] 단계 704에서, 샘플들을 갖는 랙이 입력 드로워를 통해 실험실 자동화 시스템 상에 로딩된다. 실험실 자동화 시스템은 샘플을 랙으로부터 검색한 후에 작업흐름 관리 층(50(b))에 샘플의 도달을 통지한다.

[0107] 단계 705에서, 작업흐름 관리 층(50(b))은 프로세스 제어 층(60(a))에게 샘플에 대해 요구되는 처리 명령어들로 명령한다. 예를 들어, 작업흐름 관리 층(50(b))은 프로세스 계획을 생성할 수 있고, 이를 프로세스 제어 층(60(a))에 제공할 수 있다.

[0108] 단계 706에서, 프로세스 제어 층(60(a))은 샘플 처리량을 최대화하면서 샘플의 턴어라운드 시간을 충족시키기 위해 처리 명령어들을 가장 잘 충족시킬 수 있는 자동화 하드웨어 서브시스템들(즉, 루트)을 판정한다. 프로세스 제어 층(60(a))이 처리 명령어들을 가장 잘 충족시킬 수 있는 자동화 서브시스템들을 판정하는 데 이용할 수 있는 정보는, 분석기들 또는 기타 기기들의 이용가능성, 그러한 기기들에서의 소모재들의 이용가능성, 다양한 기기들에서의 백업 등을 포함할 수 있다.

[0109] 단계 707에서, 프로세스 제어 층(60(a))은 중간 제어 층(70(a))에 샘플에 대한 전달 명령어들을 제공한다.

[0110] 단계 708에서, 중간 제어 층(70(a))은 전달 명령어들을 디바이스 커맨드들에 맵핑하고, 커맨드들을 디바이스 제어 층(70(b))으로 전송한다.

[0111] 단계 709에서, 디바이스 제어 층(70(b))은 커맨드들을 실행하고, 하드웨어를 제어하며, 커맨드 결과들(예컨대, 액추에이터 위치들, 센서 데이터 등)로 중간 제어 층(70(a))에 응답한다.

[0112] 단계 710에서, 중간 제어 층(70(a))은 샘플의 튜브 상태 데이터를 업데이트하고, 데이터를 프로세스 제어 층(60(a))으로 포워드한다. 이러한 튜브 상태 데이터는 작업흐름 관리 층(50(b))으로부터 수신된 후속 프로세스 계획들에서의 루트들을 최적화하도록 프로세스 제어 층(60(a))에 의해 이용될 수 있다.

[0113] 단계 711에서, 프로세스 제어 층(60(a))은 튜브 상태 데이터를 작업흐름 관리 층(50(b))으로 포워드한다.

[0114] 단계 711 이후에, 샘플 컨테이너는 LAS(80)에 접속 또는 비-접속될 수 있는 분석기(60(b), 60(c))에 제공될 수 있다.

[0115] 단계 712에서, 접속된 분석기들(60(b))에 대해, 샘플들이 분석기들로 직접 전달된다. 독립형 분석기들(60(c))의 경우에, 샘플들이 트레이들로 전달된다. 실험실 직원은 이들 트레이들을 언로딩하고, 샘플들을 적절한 분석기들로 수동으로 로딩한다.

[0116] 단계 713에서, 하나 이상의 분석기들(60(b), 60(c))이 샘플을 처리한다. 분석기들(60(b), 60(c))은 테스트 결과들을 작업흐름 관리 층(50(b))으로 전송한다.

[0117] 단계 714에서, 작업흐름 관리 층(50(b))은 테스트 결과들을 입증한다.

[0118] 단계 715에서, 일단 입증되면, 작업흐름 관리 층(50(b))은 테스트 결과들을 실험실 정보 시스템(50(a))으로 포워드하고 샘플 완성 메시지를 프로세스 제어 층(60(a))으로 포워드하여 완성된 샘플이 저장될 수 있게 한다.

[0119] 단계 716에서, 실험실 정보 시스템(50(a))은 입증된 테스트 결과들을 의료 또는 실험실 직원에게 방출한다.

[0120] **그래픽 사용자 인터페이스(GUI)**

[0121] 관리 아키텍처(100)에 대한 그래픽 사용자 인터페이스(GUI)는 LAS 시스템에 대한 공통 콘솔을 사용할 수 있다. 대부분의 기기들은 프레임-GUI들로 지칭되는 부착된 대응하는 그래픽 디스플레이를 가질 수 있는데, 그 이유는 그들이 하드웨어 프레임에 물리적으로 부착되기 때문이다.

[0122] GUI 개념은 위에서 상세히 설명된 레벨 개념에 적응한다. 각각의 레벨(실험실 레벨(50), 기기 레벨(60), 및 서브시스템 레벨(70))에 대해, 소정의 기능 범위가 유용성 및 안전성, 우수한 조작처리, 유연성, 및 통합 능력을 확인하는 데 이용될 수 있다. 안정성 사안들에 대해, 사용자 권리들이 인터페이스들의 구성에 대한 액세스를 한정한다. 상측 레벨에서, GUI는 한정된 정보를 실험실 개관에 제공하여 임의의 변화에 반응하고 자발적인 작

용들을 가능하게 할 수 있다. 다른 레벨들은 실험실-기기-서브시스템 계층 내의 그들의 위치에 따라 구성된다.

[0123] 랩톱과 같은 모바일 디바이스로부터 임의의 서브시스템, 기기, 또는 중앙 실험실 개관에 액세스할 가능성을 가짐으로써 특수한 역할이 서비스 레벨에 할당될 수 있다. 상이한 레벨들이 도 3에 도시되어 있다. 도 3에 도시된 바와 같이, 실험실 레벨(502)은 실험실 레벨(502(b)) 또는 디스플레이 탭 정보(502(c))에 드릴 다운(drill down)할 수 있는 실험실 개관 부분(502(a))을 포함할 수 있다. 실험실 레벨(502(b))은 또한 실험실 구성(502(d))에 액세스할 수 있다. 실험실 개관(502(a))은 또한 기기 레벨(504) 내의 기기 레벨 인터페이스(504(b))에 액세스할 수 있다. 기기 레벨 인터페이스(504(b))는 기기 개관 인터페이스(504(a))에 액세스할 수 있고, 이 기기 개관 인터페이스(504(a))는 기기 구성 인터페이스(504(c)) 및 서브시스템 레벨(506) 내의 서브시스템 레벨 인터페이스(506(a))에 액세스할 수 있다. 서브시스템 레벨 인터페이스(506(a))는 서브시스템 구성 인터페이스(506(b))에 액세스할 수 있다.

[0124] 실험실 레벨(502)에서, GUI 아키텍처는 사용자 인터페이스를 도울 수 있는 적어도 두 개의 설계 요소들에 실험실 레벨(502)에 있는 시스템을 제공할 수 있다. 그들은 TV 스크린 클라이언트 및 재검토 클라이언트를 포함할 수 있다.

[0125] 이러한 레벨(502)만이 제어 층들에 대한 사용자 레벨 액세스를 제공한다. TV 스크린 클라이언트는 WML(50(b))에 접속된 디스플레이 전용 디바이스일 수 있다. 그것은 WML(50(b))에 접속된 기기들의 상태, 관심 있는 것들만을 디스플레이하도록 필터링될 수 있는 샘플 상태, 및 기기 에러 메시지들과 같은 실험실의 개관을 보여줄 수 있다.

[0126] 재검토 클라이언트는 제어 층들의 모든 액세스 레벨들에 대해 개방적일 수 있다. 재검토 클라이언트는 실험실의 실험실 레벨 뷰에 대한 액세스를 허용한다. 재검토 클라이언트로부터, TV 클라이언트를 구성하는 것이 가능하다. 또한, 에러 메시지들 또는 기타 경고들, 그리고 WML(50(b))에 접속된 각각의 기기의 상태 및 상황이 여기서 관찰될 수 있다. 그것은 실험실 환경을 설정하고 물리적 기기들에 대한 링크들을 제공하는 구성 마법사를 가질 수 있다. 여기서, 사용자 인터페이스는 기기 및 서브시스템 레벨들에서의 상태를 관찰하도록 드릴 다운할 수 있다.

[0127] 실험실 레벨(502)이 모든 접속된 및 비-접속된 기기들 그리고 WML(50(b))에 접속된 LAS와 관련되지만, 기기 레벨(504)은 일부 실시 형태들에서 오로지 하나의 특정 LAS와만 관련될 수 있다. GUI 아키텍처는 사용자 인터페이스를 돕는 두 개의 설계 요소들인 기기 클라이언트 및 서비스 클라이언트에 기기 레벨(60)에 있는 시스템을 제공한다.

[0128] 기기 클라이언트는 각각의 사용자 액세스 레벨이 기기 레벨(504)로부터의 LAS를 관리할 수 있게 한다. 이러한 클라이언트는 PCL에 대한 GUI로서 사용된다. 기기 클라이언트는 PCL에 대한 원격 사용자 인터페이스로서 작용하여 사용자가 상태를 관찰하고 구성 변화들을 행할 수 있게 한다. PCL은 에러들, 서브시스템 상태, 소모재들 등에 관한 상태 정보를 기기 클라이언트에 제공할 필요가 있고, 기기 클라이언트는 서브시스템들을 정지시키고, 에러들로부터 복구하고, 그리고 구성 설정들을 수정하도록 커맨드들을 PCL로 전송할 수 있다. 일부 경우에, PCL 및 기기 클라이언트는 동일한 컴퓨터형 장치 상에 존재할 수 있지만, 상이한 프로세스들로 작동될 수 있다.

[0129] 이러한 레벨은 서브시스템 레벨(506)의 판독 전용 뷰들을 관찰하도록 내비게이트 다운(navigating down)하는 것을 허용한다. 그것은 또한 LAS의 일부인 특정 서브시스템들에 관련된 임의의 기기 에러 메시지들 및 디스플레이 상태 정보를 허용한다. 그것은 또한 기기의 일상적인 동작들을 수행하고, 기기 레벨(504)에서의 에러 복구, 기기 레벨 초기화, 기기 구성 작업들, 그리고 백업 및 복원 동작들을 수행한다.

[0130] 서비스 클라이언트는 서비스 및 최고 사용자 액세스를 제공하여 LAS의 각각의 레벨에 걸친 기능들을 서비스한다. 서비스 클라이언트는 서비스 랩톱 또는 다른 컴퓨터형 디바이스 상에서 작동할 수 있다.

[0131] 이러한 레벨은 서브시스템 레벨(506)의 판독 전용 뷰들을 관찰하도록 내비게이트 다운하는 것을 허용한다. 그것은 또한 LAS의 일부인 특정 서브시스템들에 관련된 임의의 기기 에러 메시지들 및 상태 정보의 디스플레이를 허용한다. 그것은 또한 기기의 일상적인 동작들, 기기 레벨(504)에서의 에러 복구, 기기 레벨 초기화, 기기 구성 작업들, 및 백업/복원 동작들을 수행한다.

[0132] 이러한 레벨은 LAS 상의 각각의 개별 서브시스템들을 책임질 것이다. 서브시스템 사용자 인터페이스는 에러가 발생하는 경우의 에러 정보와 함께 서브시스템의 상태를, 그리고 가능하게는 복구 프로세스를 돕는 에러 복구 비디오들을 보여줄 수 있다. 사용자들은 서브시스템 사용자 인터페이스를 사용하여, 예를 들어 입력 또는 출력

에서 활성 베이스 프레임 구성을 수정하여, 일부 기본 서브시스템 구성 작업들을 수행할 수 있다.

[0133] GUI 아키텍처는 사용자 인터페이스를 돕는 두 개의 설계 요소들인 프레임 클라이언트 및 서비스 클라이언트에 서브시스템 레벨에 있는 시스템을 제공한다.

[0134] 프레임 클라이언트는 디스플레이될 것을 요청받은 특정 서브시스템에 대해 특정적이다. 각각의 프레임 클라이언트에 대해, 모든 프레임 클라이언트들 사이에서 공통적인 GUI의 일부가 있을 수 있다. 이러한 공통성은 공통 프레임 클라이언트 내에서 그룹화될 수 있고, 비공통 부분들은 서브시스템 특정 프레임 클라이언트의 일부일 것이다.

[0135] 프레임 클라이언트는 현재 서브시스템, 현재 서브시스템 상태, 경고 조건들, 및 사용자가 주목할 필요가 있는 서브시스템 상의 위치를 디스플레이할 수 있다. 프레임 클라이언트는 또한 서브시스템 상의 튜브들의 공급물 또는 폐기물 상태를 디스플레이할 수 있다.

[0136] 서비스 클라이언트는 서비스 또는 관리 사용자가 하기의 서비스 유형 기능들을 수행하게 할 수 있다: (1) 기기 커맨드 기능들 - 서브시스템의 시작, 정지, 서브시스템의 초기화; (2) 드로워들의 개방/폐쇄; (3) 랙들의 교환; (4) 공급물 교체 또는 폐기물 제거; (5) 에러 복구의 수행; (6) 서브시스템 구성의 수행; (7) 시스템 교시 기능들의 수행; (8) 정렬 절차들; (9) 백업/복원 기능들; (10) 서브시스템에 대한 보수 절차들; (11) 진단 특징부들; (12) 시스템 실행기 기능들; 및 (13) 성능 검증 테스트들.

[0137] 관리 아키텍처 시스템에서, 에러들 및 동작 이벤트들이 "이벤트들"로서 일반화된다. 관리 아키텍처는 - 이벤트들이 시스템 에러들인지 아니면 동작 이벤트들인지와는 무관하게 - 이벤트들을 다루는 데 있어서 단일 메커니즘인 이벤트 경고 객체(EAO)를 채용한다.

[0138] LAS 또는 분석기가 조우하는 이벤트들 중 일부는 실제로 에러들(예컨대, 로봇 움직임 에러, 고정된 캐리어 등)인 반면, 일부 이벤트들은 동작 이벤트들이고, 시스템의 정상 동작(예컨대, 디캡퍼 폐기물의 처분, 전체 트레이들의 언로딩, 분취 선단들의 재충전)의 일부이고 에러가 아니다. 동작 이벤트들은 사용자가 이벤트를 통지받을 필요가 있도록 에러와 유사하게 거동한다. 사용자는 동작 이벤트로부터 복구하도록 추가적인 명령어들을 요구할 수 있다. 일단 동작 이벤트로부터의 복구가 완료되면, GUI 상에 디스플레이되는 대응하는 통지들이 제거될 필요가 있다.

[0139] 이벤트 경고 객체는 이벤트 또는 경고 데이터를 포함하고, 실험실, 기기, 및 서브시스템 레벨들 사이의 수송 메커니즘으로서 이용된다. 그것은 경고가 디스플레이될 레벨, 및 에러 복구가 수행될 장소를 명시할 것이다. 하기의 섹션들은 관리 아키텍처의 각각의 아키텍처 레벨에 대한 에러 복구의 거동을 기술한다.

[0140] 이러한 레벨에서, 모든 실험실 또는 샘플 이벤트들이 관찰될 수 있다. 이러한 레벨에서 디스플레이되는 이벤트/에러 통지들은 에러 복구를 수행할 레벨을 나타낼 것이다. 샘플 기반 에러들에 대해, WML은 튜브에 대한 처리가 에러 또는 이벤트의 결과로서 논의 중인 경우에 프로세스 계획을 생성하여 샘플을 에러 위치에 잠재적으로 전달할 수 있다. 또한, WML에 새로운 프로세스 계획을 생성할 것을 요구하는 에러 조건들이 WML로 포워드될 것이다. 이어서, 새로운 프로세스 계획이 기기 레벨에 이르기까지 전송될 것이다.

[0141] 비-샘플 기반 에러들의 경우에, WML의 부전(failure) 또는 WML과 기기 레벨 사이의 통신의 부전은 샘플들로 하여금 그들이 있는 장소에서 유지되게 하거나 또는 그들이 작동 중인 분석기에서의 처리를 완료하게 하고, 이어서 WML과의 접속이 재확립될 때까지 현재 서브시스템에서 유지되게 할 것이다. 접속이 재확립되지 않는 경우에(즉, 약간의 주어진 휴식기 후에), 기기 레벨은 가능하다면, 서브시스템 레벨에게 루트 레그들을 완료할 것을 지시할 수 있고, 그렇지 않다면, 튜브들을 에러 위치로 전달할 것을 지시할 수 있다.

[0142] 자동화 특정 이벤트들이 이러한 레벨로부터 관찰가능할 것이다. 이러한 레벨에서 디스플레이되는 이벤트/에러 통지들은 에러 복구를 수행할 수 있는 레벨을 나타낼 수 있다.

[0143] 샘플 기반 에러들의 경우에, WML이 현재 프로세스 계획을 충족시키는 것을 방해하는 에러 또는 이벤트가 발생했음을 나타내도록 튜브 상태 정보 메시지가 WML로 전송될 수 있다. 또한, WML은 이어서 새로운 프로세스 계획을 전송하여 튜브를 에러 위치 또는 일부 다른 새로운 목적지 중 어느 하나로 전달할 것이다. 기기 레벨은 또한 프로세스 계획이 생성될 것을 요구하지 않는 에러들에 대해 서브시스템 레벨에 의해 생성된 에러 통지들을 디스플레이할 수 있다.

[0144] 비-샘플 기반 에러들의 경우에, 기기 레벨의 부전 또는 기기 레벨과 서브시스템 레벨 사이의 통신의 부전. 샘플은 현재 서브시스템에서의 현재 처리를 완료할 것이다. 일단 접속이 재확립되면, 튜브 상태 정보가 이를 WML

상에 포워드하는 기기 레벨로 전송된다. 현재 프로세스 계획이 만족스러운 경우에 사용되거나 또는 새로운 프로세스 계획이 생성된다. 이어서, 프로세스 계획이 정상으로서 실행된다. 접속이 재확립되지 않는다면, 샘플이 현재 서브시스템에서 대기한다.

[0145] 서브시스템에 특정적인 이벤트들이 서브시스템 레벨에서 관찰가능할 것이다. 이러한 레벨에서 디스플레이되는 이벤트/에러 통지들은 에러 복구를 수행할 레벨을 나타낼 것이다.

[0146] 샘플 기반 에러들의 경우에, MCL이 샘플 튜브가 전달되는 것을 방해하는 에러를 샘플 튜브에서 검출하면(그 이유는 MCL이 샘플 튜브의 전달을 몇몇 이유로 충족시킬 수 없기 때문이거나, 또는 튜브가 알려져 있지 않거나 예상되지 않는 위치에 있기 때문임), 그것은 기기 레벨을 통해 WML에 전송될 튜브 상태 정보 메시지를 생성할 것이다. 이어서, WML은 새로운 프로세스 계획을 생성하여 튜브를 새로운 목적지 또는 에러 위치 중 어느 하나로 전달할 수 있다. 샘플의 전달을 수반하지 않는 에러가 있다면, 서브시스템 레벨은 기기 레벨에게 통지할 것인데, 기기 레벨은 이어서 사용자에게 통지할 것이다. 기기 레벨은 또한 필요하다면 LAS를 중지시킬 수 있다.

[0147] 비-샘플 기반 에러들의 경우에, 서브시스템이 재시작하면, 그것은 캐싱된(cached) 메시지들을 기기 레벨로 전송할 것이다. 알려진 상태들을 갖는 샘플들이 처리될 수 있다. 기기 레벨은 각각의 샘플에 대한 튜브 상태 정보 메시지를 WML로 전송할 수 있다. 새로운 프로세스 계획이 수신된다면, 그것은 처리될 수 있다. 그렇지 않다면, 현재 프로세스 계획이 이용될 수 있다. 또한, 서브시스템 클라이언트는 샘플을 계속해서 처리할 것인지 아니면 그것을 SIQ(sample in question)로 전송할 것인지에 대해 사용자에게 프롬프팅할 수 있다. 처리를 계속하는 것은 성공적인 완료를 나타내는 튜브 상태 정보 메시지를 생성할 것이다. 그것을 SIQ로 전송하는 것은 에러들이 있는 완료를 나타내는 튜브 상태 정보 메시지를 전송할 것이다. 샘플이 임의의 이유에 의해 SIQ 버퍼로 전송되면, 그 이유, 예를 들어 "원심분리하기에 너무 작은 부피" 또는 "판독할 수 없는 바코드" 등을 나타내는 메시지가 WML로 전송된다.

[0148] 도 4는 본 발명의 일부 실시 형태들에 따른 방법 또는 동작을 실행하도록 구성된 컴퓨팅 장치 내에 존재할 수 있는 요소들의 블록 다이어그램이다. 도 4의 요소들은 도 1a, 도 1b, 도 1c 등에 도시된 컴포넌트들 중 임의의 것에 이용될 수 있다. 도 4에 도시된 서브시스템들은 시스템 버스(575)를 통해 상호접속된다. 프린터(574), 키보드(578), 고정식 디스크(579), 디스플레이 어댑터(582)에 연결되는 모니터(576), 등과 같은 부가의 서브시스템들이 도시된다. 입력/출력(I/O) 제어기(571)에 연결하는 주변 장치 및 I/O 디바이스들은 시리얼 포트(577)와 같은 본 기술 분야에서 알려진 임의의 수의 수단에 의해 컴퓨팅 시스템에 접속될 수 있다. 예를 들어, 시리얼 포트(577) 또는 외부 인터페이스(581)가 인터넷과 같은 광역 통신망, 마우스 입력 디바이스, 또는 스캐너에 컴퓨팅 디바이스를 접속시키는데 사용될 수 있다. 시스템 버스(575)를 통한 상호 접속은 프로그래밍된 중앙 프로세서(573)(예컨대, 마이크로프로세서, CPU, 등)가 각각의 서브시스템과 통신하게 하고, 서브시스템들 사이의 정보의 교환뿐 아니라, 시스템 메모리(572) 또는 고정식 디스크(579) 내에 저장될 수 있는 명령어들의 실행을 제어하게 한다. 시스템 메모리(572) 및/또는 고정식 디스크(579)는 컴퓨터 판독가능 매체를 구체화할 수 있다.

[0149] 본 출원에 기술된 소프트웨어 컴포넌트들 또는 기능들의 어떤 것도 예컨대, 종래의 또는 객체-지향 기술들을 사용하는 예를 들어, 자바, C++ 또는 펄(Perl)과 같은 임의의 적합한 컴퓨터 언어를 사용하는 프로세서에 의해 실행될 소프트웨어 코드로서 구현될 수 있다. 소프트웨어 코드는 랜덤 액세스 메모리(RAM), 판독 전용 메모리(ROM), 하드 드라이브 또는 플로피 디스크와 같은 자기 매체, 또는 CD-ROM과 같은 광학 매체와 같은 컴퓨터 판독가능 매체 상에 일련의 명령어들, 또는 커맨드들로서 저장될 수 있다. 임의의 그러한 컴퓨터 판독가능 매체는 단일의 컴퓨터형 장치 상에 또는 그 내에 있을 수 있고, 시스템 또는 네트워크 내의 상이한 컴퓨터형 장치들 상에 또는 그 내에 존재될 수 있다.

[0150] 상기의 설명은 예시적인 것이며, 제한적인 것이 아니다. 본 발명의 많은 변형들이 본 개시 내용의 검토 시에 당업자에게는 명백해질 것이다. 그러므로, 본 발명의 범주는 상기의 설명을 참조하여 결정되어서는 아니되며, 대신에, 그의 전체 범주 또는 등가물과 함께 첨부된 특허청구범위를 참조하여 결정되어야 한다.

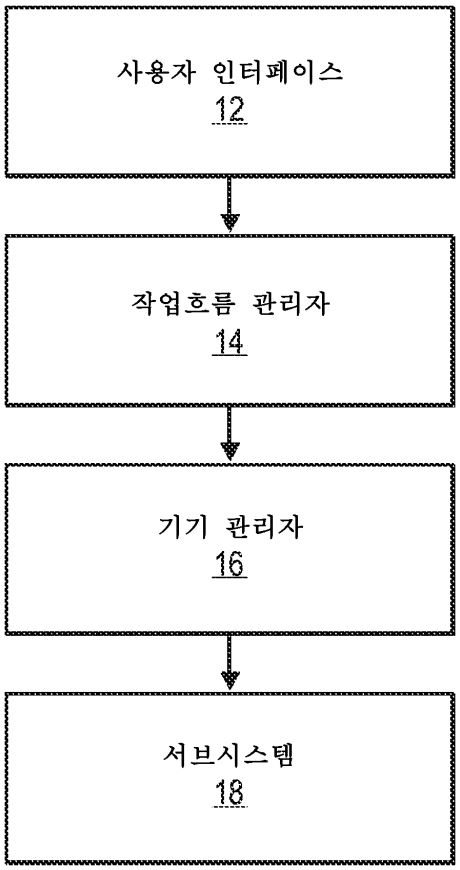
[0151] 임의의 실시 형태로부터의 하나 이상의 특징부들이 본 발명의 범주로부터 벗어남이 없이 임의의 다른 실시 형태의 하나 이상의 특징부들과 조합될 수 있다.

[0152] 단수형("a", "an" 또는 "the")의 열거는 특별히 반대로 지시되지 않으면 "하나 이상"을 의미하도록 의도된다.

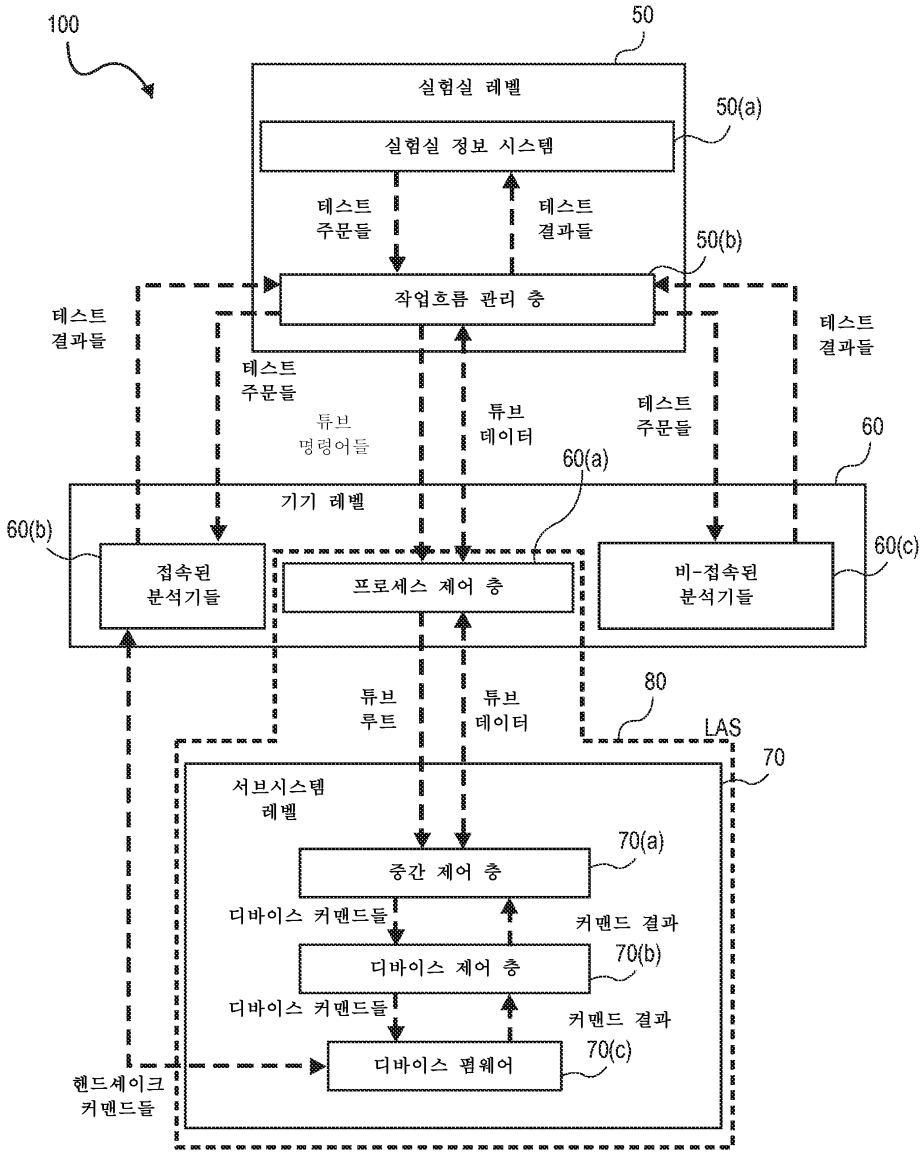
[0153] 전술한 모든 특허들, 특허 출원들, 공보들, 및 설명들은 모든 목적들을 위해 전체로서 본 명세서에 참조로 포함된다. 어느 것도 종래 기술인 것으로 인정되지 않는다.

도면

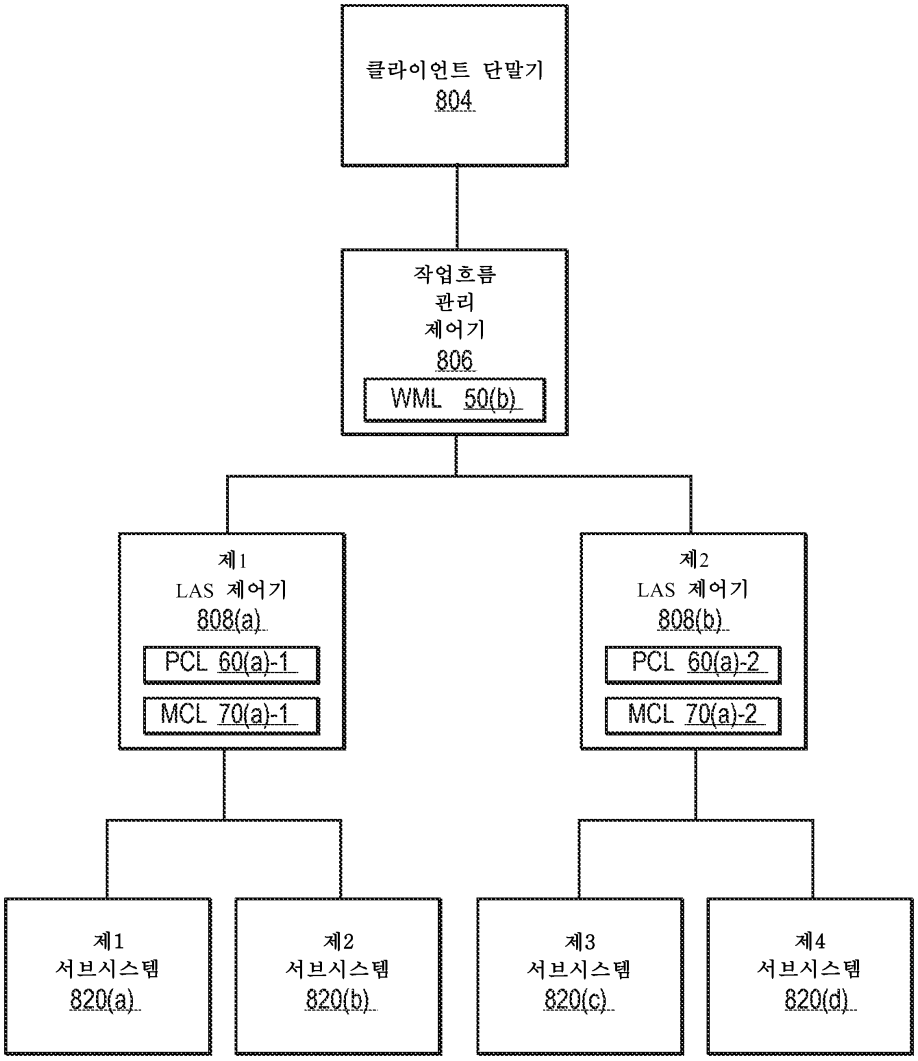
도면1a



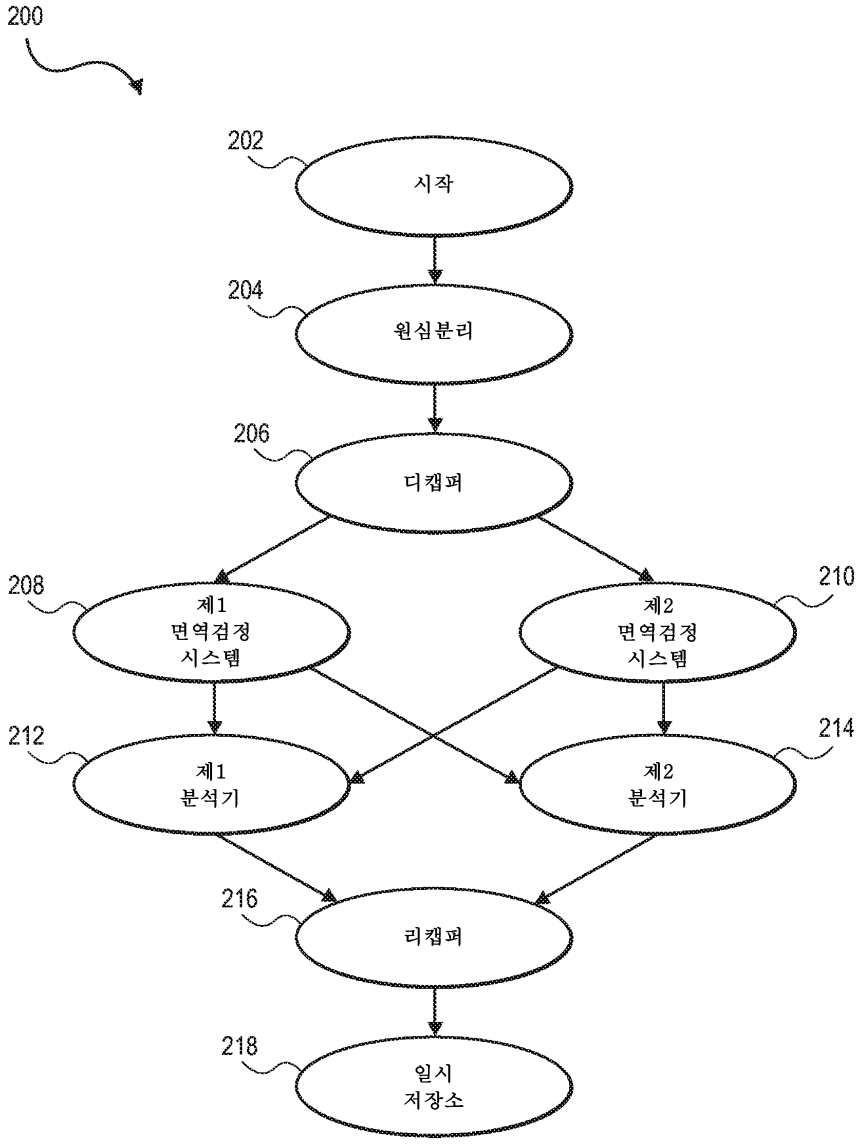
도면1b


100
50
실험실 레벨
50(a)
실험실 정보 시스템
테스트 주문들
테스트 결과들
50(b)
작업흐름 관리 층
테스트 결과들
테스트 주문들
튜브 명령어들
튜브 데이터
테스트 주문들
테스트 결과들
60
60(a)
기기 레벨
60(b)
접속된 분석기들
프로세스 제어 층
비-접속된 분석기들
60(c)
튜브 루트
튜브 데이터
80
LAS
70
서브시스템 레벨
70(a)
중간 제어 층
디바이스 커맨드들
커맨드 결과
70(b)
디바이스 제어 층
디바이스 커맨드들
커맨드 결과
70(c)
디바이스 펌웨어
핸드셰이크 커맨드들

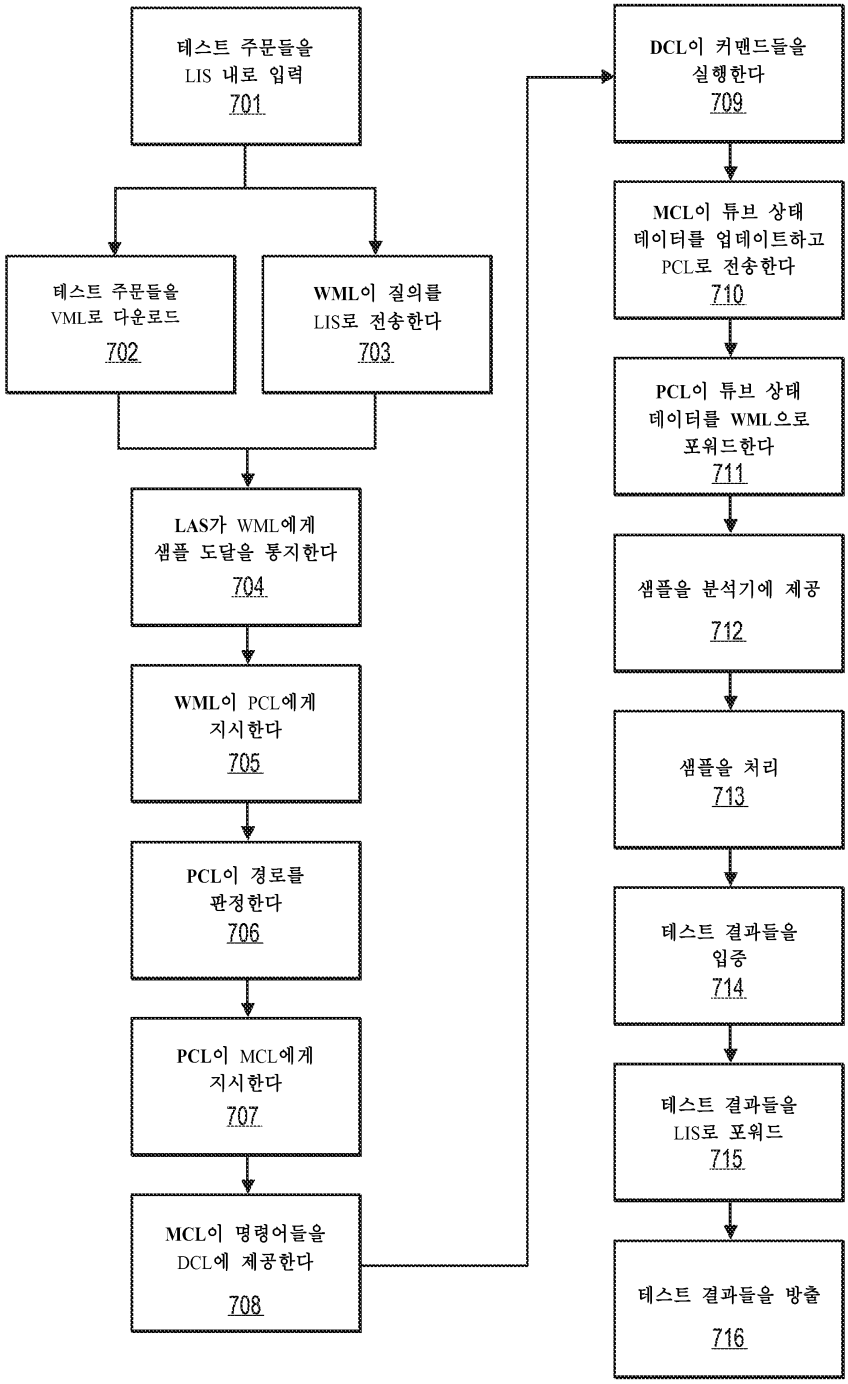
도면1c


클라이언트 단말기
804
작업흐름
관리
제어기
806
WML 50(b)
제1
LAS 제어기
808(a)
PCL 60(a)-1
MCL 70(a)-1
제2
LAS 제어기
808(b)
PCL 60(a)-2
MCL 70(a)-2
제1
서브시스템
820(a)
제2
서브시스템
820(b)
제3
서브시스템
820(c)
제4
서브시스템
820(d)

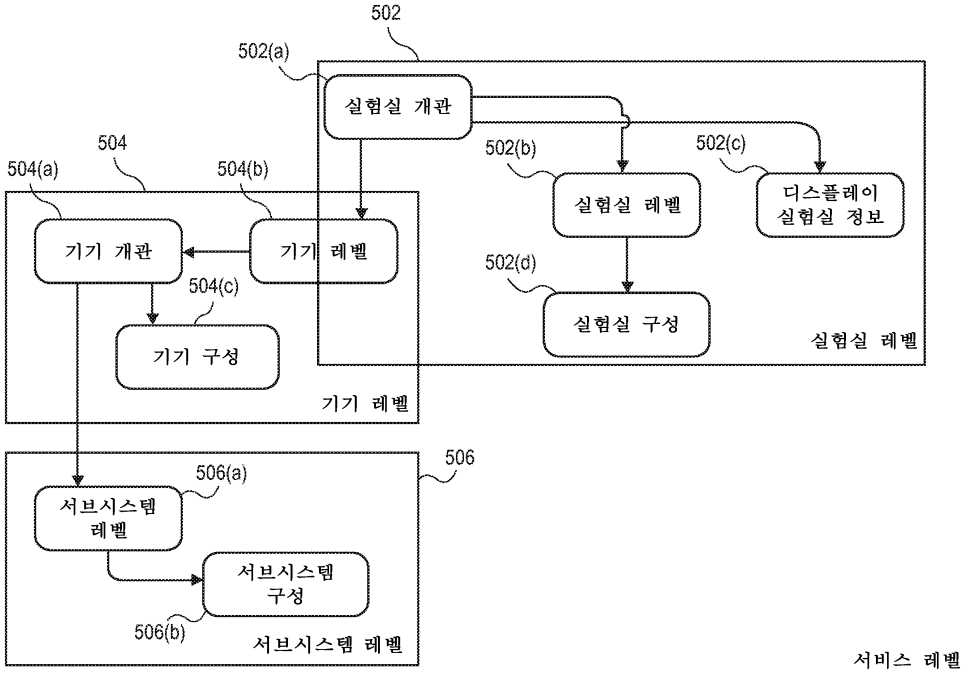
도면2a


200
202
시작
204
원심분리
206
디캡퍼
208
제1
면역검정
시스템
210
제2
면역검정
시스템
212
제1
분석기
214
제2
분석기
216
리캡퍼
218
일시
저장소

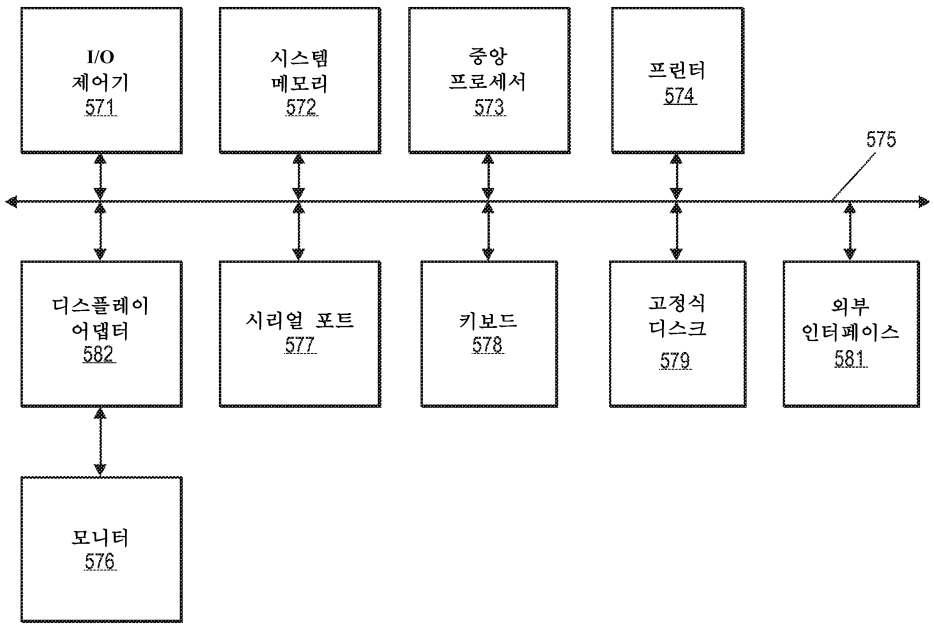
도면2b


테스트 주문들을
LIS 내로 입력
701
테스트 주문들을
VML로 다운로드
702
WML이 질의를
LIS로 전송한다
703
LAS가 WML에게
샘플 도달을 통지한다
704
WML이 PCL에게
지시한다
705
PCL이 경로를
판정한다
706
PCL이 MCL에게
지시한다
707
MCL이 명령어들을
DCL에 제공한다
708
DCL이 커맨드들을
실행한다
709
MCL이 튜브 상태
데이터를 업데이트하고
PCL로 전송한다
710
PCL이 튜브 상태
데이터를 WML으로
포워드한다
711
샘플을 분석기에 제공
712
샘플을 처리
713
테스트 결과들을
입증
714
테스트 결과들을
LIS로 포워드
715
테스트 결과들을 방출
716

도면3


502
502(a)
실험실 개관
502(b)
실험실 레벨
502(c)
디스플레이
실험실 정보
502(d)
실험실 구성
실험실 레벨
504
504(a)
504(b)
기기 개관
기기 레벨
504(c)
기기 구성
기기 레벨
506
506(a)
서브시스템
레벨
서브시스템
구성
506(b)
서브시스템 레벨
서비스 레벨

도면4


I/O
제어기
571
시스템
메모리
572
중앙
프로세서
573
프린터
574
575
디스플레이
어댑터
582
시리얼 포트
577
키보드
578
고정식
디스크
579
외부
인터페이스
581
모니터
576