(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년10월08일
(11) 등록번호 10-0920595
(24) 등록일자 2009년09월29일

(51) Int. Cl.
H04L 29/06 (2006.01)
(21) 출원번호 10-2006-7017210
(22) 출원일자 2005년01월24일
심사청구일자 2008년01월31일
(85) 번역문제출일자 2006년08월25일
(65) 공개번호 10-2007-0001973
(43) 공개일자 2007년01월04일
(86) 국제출원번호 PCT/EP2005/050299
(87) 국제공개번호 WO 2005/083967
국제공개일자 2005년09월09일
(30) 우선권주장
10/789,573 2004년02월27일 미국(US)
(56) 선행기술조사문헌
US6212160 B1
US5951639 A
EP1292160 A
EP1370044 A
전체 청구항 수 : 총 10 항

(73) 특허권자
인터내셔널 비지네스 머신즈 코포레이션
미국 10504 뉴욕주 아몬크 뉴오차드 로드
(72) 발명자
버카트 에릭
미국 27604 노스캐롤라이나주 랄리 페르시몬 릿지 드라이브 2348
폰테스 스티븐
미국 27560 노스캐롤라이나주 모리스빌 윈터 워크 서클 1336
란젠 크라이그
미국 48144 미시간주 람버트빌 렌녹스 코트 3157
(74) 대리인
윤여원, 허정훈

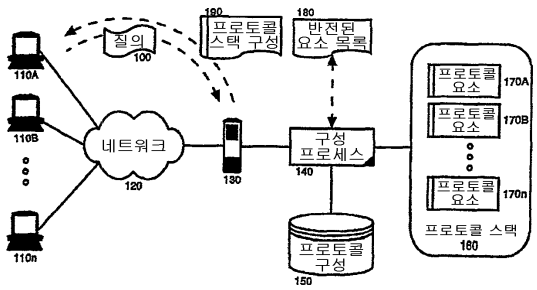
심사관 : 오제욱

(54) 클라이언트 측 통신 프로토콜 스택 구성 방법, 프로세스 간통신 프로토콜 구성 시스템 및 머신 판독가능한 저장 장치

(57) 요 약

본 발명은 클라이언트 측 통신 프로토콜 스택을 구성하는 방법 및 시스템에 관한 것이다. 한 방법에서, 서버 측 프로토콜 스택 인스턴스 내의 프로토콜 스택 요소 세트에 대응하는 클라이언트 측 프로토콜 스택 요소 세트를 결정하는 데 매핑이 참조될 수 있다. 이어서, 결정된 클라이언트 측 프로토콜 스택 요소 세트의 목록이 생성될 수 있다. 최종적으로, 외부에 배치된 클라이언트 연산 프로세스에 의한 액세스를 위해 이 목록이 공표될 수 있다. 바람직하게, 이 목록은 반전될 수 있다. 또한, 참조하는 단계는 클라이언트 측 프로토콜 스택이 서버 측 프로토콜 스택 인스턴스와 상호 연동하는 것이 가능할 때 적어도 하나의 클라이언트 측 프로토콜 스택 요소에 적용되는 적어도 하나의 속성을 결정하는 단계를 더 포함할 수 있다.

대 표 도



특허청구의 범위

청구항 1

클라이언트 측 통신 프로토콜 스택을 구성하는 방법에 있어서,

서버 측 프로토콜 스택 인스턴스(server side protocol stack instance) 내의 프로토콜 스택 요소 세트에 대응하는 클라이언트 측 프로토콜 스택 요소 세트를 결정하는 단계(220)와,

상기 결정된 클라이언트 측 프로토콜 스택 요소 세트(160)의 목록을 생성하는 단계(370)와,

외부에 배치된(disposed) 클라이언트 연산 프로세스에 의한 액세스를 위해 상기 목록을 공표하는 단계와,

상기 클라이언트 연산 프로세스(110) 중 하나에 의해 상기 클라이언트 측 프로토콜 스택 요소 세트의 목록을 요청 및 액세스하는 단계(410)와,

상기 목록으로부터 상기 요청된 클라이언트 측 프로토콜 스택 요소 세트를 포함하는 클라이언트 측 프로토콜 스택 인스턴스가 이미 존재하는지 여부를 판단하는 단계(420)와,

상기 클라이언트 측 프로토콜 스택 인스턴스가 이미 존재하면, 상기 클라이언트 측 프로토콜 스택 인스턴스를 재사용하여 상기 서버 측 프로토콜 스택 인스턴스와 통신하는 단계(440)와,

상기 클라이언트 측 프로토콜 스택 인스턴스가 존재하지 않으면, 상기 요청된 클라이언트 측 프로토콜 스택 요소 세트를 사용하여 새로운 클라이언트 측 프로토콜 스택 인스턴스를 생성하는 단계(450)를 포함하는

클라이언트 측 통신 프로토콜 스택 구성 방법.

청구항 2

제 1 항에 있어서,

상기 결정하는 단계는, 클라이언트 측 프로토콜 스택이 상기 서버 측 프로토콜 스택 인스턴스와 상호 연동하는 것이 가능하면 상기 클라이언트 측 프로토콜 스택 요소 세트 중 적어도 하나에 적용되는 적어도 하나의 속성을 결정하는 단계를 더 포함하는

클라이언트 측 통신 프로토콜 스택 구성 방법.

청구항 3

제 1 항에 있어서,

상기 결정된 클라이언트 측 프로토콜 스택 요소 세트의 목록을 생성하는 단계는 상기 생성된 목록의 순서를 반대로 하는 단계를 더 포함하는

클라이언트 측 통신 프로토콜 스택 구성 방법.

청구항 4

제 1 항에 있어서,

상기 액세스하는 단계는 기본 디렉토리 인터페이스를 통해 상기 목록에 액세스하는 단계를 포함하는

클라이언트 측 통신 프로토콜 스택 구성 방법.

청구항 5

제 1 항에 있어서,

상기 요청된 클라이언트 측 프로토콜 스택 요소 세트에 적용되는 속성을 식별하는 단계와,

상기 새로운 클라이언트 측 프로토콜 스택 인스턴스 내의 상기 클라이언트 측 프로토콜 스택 요소 세트에 상기 식별된 속성을 적용하는 단계를 더 포함하는

클라이언트 측 통신 프로토콜 스택 구성 방법.

청구항 6

제 5 항에 있어서,

상기 클라이언트 측 프로토콜 스택 인스턴스를 재사용하여 상기 서버 측 프로토콜 스택 인스턴스와 통신하는 단계는,

상기 클라이언트 측 프로토콜 스택 인스턴스가 상기 식별된 속성을 포함하는 경우에만 상기 클라이언트 측 프로토콜 스택 인스턴스를 재사용하여 상기 서버 측 프로토콜 스택 인스턴스와 통신하는 단계와,

상기 클라이언트 측 프로토콜 스택 인스턴스가 상기 식별된 속성을 포함하지 않으면, 상기 요청된 클라이언트 측 프로토콜 스택 요소 세트를 사용하여 상기 식별된 속성을 포함하는 새로운 클라이언트 측 프로토콜 스택 인스턴스를 생성하는 단계를 포함하는

클라이언트 측 통신 프로토콜 스택 구성 방법.

청구항 7

프로세스 간 통신 프로토콜 구성 시스템에 있어서,

서버 측 프로토콜 스택 인스턴스 내의 프로토콜 스택 요소 세트에 대응하는 클라이언트 측 프로토콜 스택 요소 세트를 결정하고, 상기 결정된 클라이언트 측 프로토콜 스택 요소 세트의 목록을 생성하며, 외부에 배치된 클라이언트 연산 프로세스에 의한 액세스를 위해 상기 목록을 공표하는 서버 프로세스(130,140)와,

상기 클라이언트 측 프로토콜 스택 요소 세트의 목록을 요청 및 액세스하고, 상기 목록으로부터 상기 요청된 클라이언트 측 프로토콜 스택 요소 세트를 포함하는 클라이언트 측 프로토콜 스택 인스턴스가 이미 존재하는지 여부를 판단하며, 상기 클라이언트 측 프로토콜 스택 인스턴스가 이미 존재하면, 상기 클라이언트 측 프로토콜 스택 인스턴스를 재사용하여 상기 서버 측 프로토콜 스택 인스턴스와 통신하고, 상기 클라이언트 측 프로토콜 스택 인스턴스가 존재하지 않으면, 상기 요청된 클라이언트 측 프로토콜 스택 요소 세트를 사용하여 새로운 클라이언트 측 프로토콜 스택 인스턴스를 생성하는 클라이언트 프로세스(110A)를 포함하는

프로세스 간 통신 프로토콜 구성 시스템.

청구항 8

제 7 항에 있어서,

상기 목록은 매핑에 기반하여 상기 서버 측 프로토콜 스택 인스턴스 내의 상기 서버 측 프로토콜 스택 요소 세트 중 종속적인 요소의 역순 목록을 포함하는 역목록(a reverse listing)인

프로세스 간 통신 프로토콜 구성 시스템.

청구항 9

제 7 항에 있어서,

상기 공표가 가능한 인터페이스는 상기 목록을 통합시키는 적어도 하나의 객체에 대한 기본 디렉토리 인터페이스인

프로세스 간 통신 프로토콜 구성 시스템.

청구항 10

클라이언트 측 통신 프로토콜 스택을 구성하는 컴퓨터 프로그램이 저장된 머신 판독가능한 저장 장치에 있어서,

상기 컴퓨터 프로그램은 서버 프로세서 및 클라이언트 프로세서에 의해 실행되는 경우 제 1 항 내지 제 6 항 중 어느 한 항의 단계들을 수행하는 머신 판독가능한 인스트럭션 세트를 포함하는

머신 판독가능한 저장 장치.

청구항 11

삭제

청구항 12

삭제

명 세 서

기 술 분 야

<1> 본 발명은 프로세스 간 통신(inter-process communication) 분야에 관한 것으로, 보다 구체적으로 서버 프로세스와 통신하는 클라이언트 프로세스 구성에 관한 것이다.

배 경 기 술

<2> 프로세스 간 통신은 2개 이상의 연산 프로세스 간의 정보 교환과 관련이 있는데, 이들 프로세스는 단일 연산 장치 내에 함께 배치되거나, 컴퓨터 통신 네트워크 주위에 서로 원격으로 배치된다. 전형적으로, 개별 연산 프로세스는 정보를 교환할 목적으로 설정된 통신 링크를 통해 서로 통신할 수 있다. 대부분의 경우에, 통신 링크를 설정하는 동안 또는 이전에 결정된 공용 통신 프로토콜을 사용하여 정보를 교환할 수 있다.

<3> 다수의 경우에, 정보 교환을 위해 2개 이상의 연산 프로세스에 의해 이용되는 공용 통신 프로토콜은 통신 링크를 설정하려는 임의의 시도가 이루어지기 전에 사전결정될 수 있다. 예컨대, 인터넷 환경에서, 웹콘텐츠는 TCP/IP를 통해 HTTP(hypertext transfer protocol)를 이용하여 콘텐츠 서버와 콘텐츠 브라우저 사이에서 교환될 수 있다. 또한, 통신 프로토콜은 반드시 물리 네트워크 층에 기반한 데이터 교환으로 한정되는 것은 아니다. 오히려, 통신 프로토콜은 2개 이상의 연산 프로세스 간의 정보 교환을 허용하는 데 필요한 통신 구성 및 실행 집합체를 포함할 수 있다. 이 점에 있어서, 통신 프로토콜은 프로세스 사이에서 데이터를 전달하는 메커니즘뿐만 아니라, 일부를 명명하기 위한 암호화, 압축과 같은 부수적인 설정과 실행 및 터널링 프로토콜도 포함할 수 있다.

<4> 어떤 경우에, 통신 프로토콜은 클라이언트 연산 프로세스와 서버 연산 프로세스 사이에서처럼 사전결정되지 않았을 것이다. 특히, 하나의 서버 연산 프로세스가 다수의 다른 통신 프로토콜을 사용하여 복수의 클라이언트 연산 프로세스와 통신하도록 구성되는 경우에, 어떤 연산 프로세스가 클라이언트와 서버 사이에서 임의의 한 세션을 이용하는지 사전결정될 수 없다. 이에 따라, 흔히 클라이언트 연산 프로세스가 프로세스의 시행착오를 통해 적합한 통신 프로토콜을 선택하게 된다. 구체적으로, 전형적인 경우에, 클라이언트 프로세스는 복수의 가능한 통신 프로토콜 표를 포함할 수 있다. 이처럼, 클라이언트 프로세스는 통신 링크가 성공적으로 설정될 때까지 각각의 가능한 통신 프로토콜을 사용하는 서버 프로세스와의 통신 링크를 설정하려 할 수 있다.

<5> 상술한 서버 프로세스 간의 통신 링크를 설정하는 프로세스는 궁극적으로 통신 링크가 알려진 통신 프로토콜 중 하나를 사용하여 설정될 수 있다면 효과적일 수 있다. 또한, 당업자는 통신 링크 설정시에 최종적으로 선택되는 통신 프로토콜이 아닌 다른 특정 통신 프로토콜을 사용하여 통신 링크를 설정하는 각각의 실패로 인해 발생할 수 있는 심한 리소스의 낭비를 알 것이다. 가능한 한, 통신 프로토콜 선택 시에 프로세스의 시행착오를 방지하여 높은 수준의 성능을 얻는 것이 바람직할 것이다.

발명의 상세한 설명

<6> 본 발명은 프로세스 간 통신에 대한 기술의 부족을 해결하고, 클라이언트 측 연산 프로세스에서 통신 프로토콜 스택을 구성하는 방법 및 시스템을 제공한다. 본 발명의 바람직한 일 측면에서, 프로세스 간 통신 프로토콜 구성 시스템은 서버 측 프로토콜 스택에서 서버 측 프로토콜 스택 요소를 클라이언트 측 프로토콜 스택 요소에 매핑하는 것을 포함할 수 있다. 이 시스템은 매핑에 기반하여 서버 측 프로토콜 스택 내의 서버 측 프로토콜 스택 요소 중 종속적인 요소의 목록을 포함할 수 있다. 최종적으로, 시스템은 목록에 대한 공표가능한 인터페이스를 포함할 수 있다.

<7> 매핑은 서버 측 프로토콜 스택의 프로토콜 스택 요소와 관련된 클라이언트 측 프로토콜 스택에 존재하도록 요청되는 대응하는 프로토콜 스택 요소 사이의 연결(linkage) 세트를 포함할 수 있다. 이 매핑은 대응하는 프로토콜 스택 요소 중 선택된 요소에 대한 속성 세트도 포함할 수 있다. 본 발명의 바람직한 측면에서, 이 목록은 매핑에 기반하여 서버 측 프로토콜 스택 내의 서버 측 프로토콜 스택 요소 중 종속적인 요소의 역순 목록을 포

함하는 역목록이 될 수 있다. 또한, 공표가능한 인터페이스는 목록을 통합시키는 적어도 하나의 객체에 대한 기본 디렉토리 인터페이스인 것이 바람직하다.

<8> 클라이언트 측 통신 프로토콜 스택을 구성하는 방법에서, 서버 측 프로토콜 스택 인스턴스 내의 프로토콜 스택 요소 세트에 대응하는 클라이언트 측 프로토콜 스택 요소 세트를 결정하는 데 매핑이 참조될 수 있다. 이어서, 결정된 클라이언트 측 프로토콜 스택 요소 세트의 목록이 생성될 수 있다. 최종적으로, 외부에 배치된 클라이언트 연산 프로세스에 의한 액세스를 위해 이 목록이 공표될 수 있다. 바람직하게, 이 목록은 반전될 수 있다. 또한, 참조하는 단계는 클라이언트 측 프로토콜 스택이 서버 측 프로토콜 스택 인스턴스와 상호 연동하는 것이 가능할 때 적어도 하나의 클라이언트 측 프로토콜 스택 요소에 적용되는 적어도 하나의 속성을 결정하는 단계를 더 포함할 수 있다.

<9> 이와 달리, 클라이언트 측 통신 프로토콜 스택을 구성하는 방법은 서버 측 프로토콜 스택 인스턴스에 모인 프로토콜 스택 요소에 대응하는 요청된 클라이언트 측 프로토콜 스택 요소의 목록을 액세스하는 단계를 포함할 수 있다. 요청된 클라이언트 측 프로토콜 스택 요소를 포함하는 클라이언트 측 프로토콜 스택 인스턴스가 이미 존재하는 지를 목록으로부터 판단할 수 있다. 만일 클라이언트 측 프로토콜 스택 인스턴스가 이미 존재하면, 서버 측 프로토콜 스택과 통신하기 위해 클라이언트 측 프로토콜 스택 인스턴스를 재사용할 수 있다. 그렇지 않으면, 요청된 클라이언트 측 프로토콜 스택 요소를 사용하여 새로운 클라이언트 측 프로토콜 스택 인스턴스가 생성될 수 있다.

<10> 이제 후속하는 도면을 예시로서만 참조하여 본 발명의 실시예를 설명할 것이다.

실 시 예

<15> 본 발명은 동적으로 선택되고 구성되는 통신 프로토콜 스택에 기반하여 서버 연산 프로세스와 통신하는 클라이언트 연산 프로세스를 지능적으로 구성하는 방법, 시스템 및 장치에 관한 것이다. 본 발명에 따르면, 프로토콜 스택 내의 상이한 해당 프로토콜 요소 또는 성분은 서버 연산 프로세스와 이 서버 연산 프로세스를 액세스하도록 구성된 하나 이상의 클라이언트 연산 프로세스 사이에서 매핑(map)될 수 있다. 예상되는 클라이언트 측 프로토콜 요소의 속성들에 매핑하는 구체적으로 지정된 속성을 갖는 서버 측 프로토콜 요소에 있어서, 이들 속성은 매핑으로도 지정될 수 있다. 지정된 속성에 따른 이 매핑은 서버 내에 프로토콜 스택을 배치할 때 다음 사용을 위해 저장될 수 있다.

<16> 프로토콜 스택이 서버 연산 프로세스 내에서 사용하기 위해 배치될 때, 프로토콜 스택에 대한 종속적인 프로토콜 요소의 목록이 생성될 수 있고, 이 목록은 순서에 있어서 반전될 수 있다. 종속적인 프로토콜 요소의 목록 내의 각 프로토콜 요소에 있어서, 임의의 특성값이 검색될 수 있다. 이에 따라, 프로토콜 스택에 대한 클라이언트 구성은 반전된 종속적인 프로토콜 요소의 목록과 검색된 프로토콜 스택의 특성 종속성에 대한 임의의 특성값을 모두 사용하여 생성될 수 있다. 이어서, 이 구성은 서버와 통신하려하는 클라이언트 프로세스가 저장된 구성을 발견할 수 있도록 공표될 수 있다.

<17> 클라이언트 연산 프로세스가 서버 연산 프로세스와 통신하려 할 때, 클라이언트 연산 프로세스는 우선 이전 공표를 통해 서버 지정형 프로토콜 구성에 질의할 수 있다. 일단 발견되면, 클라이언트 프로세스는 지정된 프로토콜 구성이 클라이언트 프로세스 내에 이미 존재하는지를 판단할 수 있다. 만일 존재하면, 프로토콜 구성은 서버 프로세스와 통신하기 위해 재사용될 수 있다. 그렇지 않으면, 지정된 구성은 서버 내에 생성될 수 있고, 이 구성의 종속성에 대한 관련 특성값으로 구성될 수 있다. 프로토콜 스택이 동작가능해 질 때, 클라이언트 프로세스는 이 프로토콜 스택을 사용하여 서버 프로세스와 통신할 수 있다.

<18> 상술한 것 중 보다 구체적인 예에서, 도 1은 본 발명에 따른 클라이언트 프로세스를 액세스하는 서버 측 구성으로 설정된 전형적이지만 비배타적(non-exclusive)인 연산 시스템의 개략도이다. 이 시스템은 컴퓨터 통신 네트워크(120)를 통해 서버 프로세스(130)와 결합하는 통신을 위해 프로그램된 하나 이상의 클라이언트 연산 프로세스(110A, 110B, 110n)를 포함할 수 있다. 특히, 각 클라이언트 연산 프로세스(110A, 110B, 110n)는 설정된 통신 프로토콜을 사용하여 서버 연산 프로세스(130)와 통신할 수 있다. 설정된 통신 프로토콜은 예컨대 TCP/IP를 사용하는 HTTP와 같은 물리 및 응용 프로토콜뿐만 아니라, 암호화 또는 압축 프로세스와 같은 하나 이상의 부수적인 절차의 실행도 포함할 수 있다.

<19> 서버 통신 프로세스(130)는 구성 프로세서(140)와 결합할 수 있다. 구성 프로세서(140)는 프로토콜 구성(150) 세트를 설정할 수 있다. 구체적으로, 구성 프로세서(140)는 서버 연산 프로세스(130)에서 프로토콜의 실행을 위해 요청되는 개별 요소를 클라이언트 연산 프로세스(110A, 110B, 110n) 중 어느 하나 내의 대응하는 요청되는

요소에 매핑할 수 있다. 프로토콜 스택이 서버 연산 프로세스(130)에서 생성될 때, 스택 내의 요소 세트의 요소 종속성은 목록에 기록되고 이어서 역구성(180)을 생성하기 위해 반전된다. 역구성(180) 내의 요소의 개별 특성값이 추가로 결정될 수 있고, 프로토콜 스택 구성(190)으로서 조합이 공표될 수 있다. 구체적으로, 종래 기술에 잘 알려져 있는 것처럼, 프로토콜 스택의 구성(190)은 기본 디렉토리 인터페이스를 통해 콘텐츠가 인식될 수 있는 하나 이상의 클래스 객체 세트에 대한 인터페이스로서 포괄적인 네임 스페이스(global names space)에서 공표될 수 있다.

<20> 실행시에, 클라이언트 연산 프로세스(110A, 110B, 110n) 중 하나가 서버 연산 프로세스(130)와의 통신 링크를 설정하기를 원할 때, 클라이언트 연산 프로세스(110A, 110B, 110n)는 우선 서버 연산 프로세스(130)가 바람직한 통신 프로토콜 구성을 식별할 것을 요청하는 질의(100)를 서버 연산 프로세스(130)에 전달할 수 있다. 이에 응답하여, 서버 연산 프로세스(130)는 클라이언트 연산 프로세스(110A, 110B, 110n) 중 요청하는 하나가 프로토콜 스택 구성(190)에 의해 정의된 바람직한 통신 프로토콜 스택을 선택하고 구성할 수 있는 프로토콜 스택 구성(190)에 대한 액세스를 반환할 수 있다.

<21> 바람직한 통신 프로토콜 구성을 사용하여 서버 연산 프로세스와 통신하는 클라이언트 연산 프로세스를 구성하는 프로세스는 서버 측 프로토콜과 클라이언트 측 프로토콜을 서로 매핑으로 연결하는 단계를 포함할 수 있다. 이 프로세스는 서버 연산 프로세스에서 통신 프로토콜 스택을 초기화하고, 매핑에 기반하여 대응하는 사용가능한 클라이언트 측 통신 프로토콜 스택 구성을 전개하는 단계도 포함할 수 있다. 최종적으로, 이 프로세스는 서버 지정형 프로토콜 스택에 기반하여 다른 클라이언트 측 프로토콜 요소의 특정 프로토콜 스택을 선택하고 구성하는 단계를 포함할 수 있다.

<22> 상술한 프로세스를 더 설명하기 위해, 도 2는 도 1의 시스템에서 서버 측 구성 내의 대응하는 서버 프로토콜 요소와 클라이언트 프로토콜 요소를 연결(link)하는 프로세스를 도시하는 흐름도이다. 블록(210)에서 처음에, 제 1 프로토콜 세트가 구성을 위해 로딩될 수 있다. 블록(220)에서, 서버 연산 프로세스에 대한 세트 내의 각 프로토콜 요소는 예상 통신 클라이언트 프로세스에 대응하는 종속적인 프로토콜 요소에 연결되거나 매핑될 수 있다. 예컨대, 서버 연산 프로세스 내의 HTTP 애플리케이션 레벨 프로토콜의 사용은 클라이언트 연산 프로세스 내의 HTTP에 매핑될 수 있다. 이와 유사하게, 서버 연산 프로세스 내의 물리 제어층 프로토콜과 같은 TCP는 클라이언트 연산 프로세스 내의 TCP에 매핑될 수 있다. 또 다른 예로서, 서버 연산 프로세스 내의 압축 해제 프로세스는 클라이언트 연산 프로세스 내의 압축 프로세스에 매핑될 수 있다.

<23> 어느 경우에도, 결정 블록(230)에서, 종속적인 프로토콜 요소 중에서 선택된 요소의 특정 속성이 서버 연산 프로세스 내의 프로토콜 요소 중 하나에 대응하는 특정 속성의 애플리케이션에 기반하여 적용되는 지를 판단할 수 있다. 만일 속성이 매핑되면, 이 속성은 매핑으로 지정될 수 있다(블록 240). 그렇지 않으면, 결정 블록(250)에서, 추가 프로토콜 세트가 프로세싱되기 위해 남아있는 지를 판단할 수 있다. 만일 남아있다면, 블록(260)에서 프로토콜 세트는 구성을 위해 로딩될 수 있고, 이 프로세스는 블록(220) 내지 블록(260)을 반복할 수 있다. 매핑으로 구성되기 위해 남아있는 프로토콜 세트가 더 이상 없다면, 이 프로세스는 블록(270)에서 종료할 수 있다.

<24> 다른 프로토콜 스택 요소 및 대응하는 속성이 매핑으로 연결되었다면, 공표된 프로토콜 스택의 특정 인스턴스에 대한 구성은 이 매핑을 사용하여 생성될 수 있다. 보다 구체적으로, 도 3은 도 1의 시스템에서 서버 내의 다른 프로토콜 요소의 프로토콜 스택을 초기화하는 프로세스를 도시하는 흐름도이다. 블록(310)에서 처음에, 프로토콜 스택이 초기화될 수 있다. 객체 지향 언어 환경에서, 예컨대, 프로토콜 스택 내에 모인 상이한 프로토콜 요소는 프로세스 주소 공간 내에서 설명되고 거기서 실행가능해질 수 있다.

<25> 블록(320)에서, 클라이언트 측 종속 요소의 목록은 예컨대, 서버 측 프로토콜 스택 내의 각 요소를 트래버싱(traverse)하고, 매핑으로 요소 종속성을 구별함으로써 생성될 수 있다. 일단 프로토콜 스택 내의 종속 요소의 목록이 생성되었다면, 블록(330)에서 목록의 순서가 반전될 수 있다. 이러한 방법으로, 이 목록에 기반하여 프로토콜 스택을 구성하는 클라이언트 연산 프로세스는 스택 내의 각 요소가 자신의 종속성에 의존할 수 있음을 보장할 수 있다. 어느 경우에도, 결정 블록(340)에서, 스택의 종속성이 이용가능한 요소 인스턴스에 의존한다는 점에서 이 종속성이 적합한 지를 판단할 수 있다. 만일 적합하지 않으면, 이 프로세스는 블록(350)에서 종료할 수 있다. 그렇지 않으면, 이 프로세스는 블록(360)까지 계속될 수 있다.

<26> 블록(360)에서, 만일 적합한 종속성이 존재하면, 각 통신 프로토콜 요소의 특성값이 검색될 수 있다. 또한, 블록(370)에서, 역순 목록에 기반하여 클라이언트 내에 배치되는 프로토콜 스택에 대한 구성이 생성될 수 있다. 최종적으로, 블록(380)에서, 검색된 특성값은 생성된 클라이언트 프로토콜 스택에 대한 구성에 첨가될 수 있다.

블록(390)에서, 요청시에 임의의 알려진 메커니즘을 통해 클라이언트 프로토콜 스택에 대한 구성이 요청하는 클라이언트 프로세스에 전달될 수 있다. 본 발명의 바람직한 측면에서, 이 구성은 기본 디렉토리 인터페이스를 통해 포괄적인 네임스페이스에서 공표될 수 있다.

<27> 일단 공표되면, 클라이언트 측 프로토콜 스택 구성은 서버 연산 프로세스가 선호하는 통신 프로토콜을 사용하는 서버 연산 프로세스와의 통신 링크 설정에 관계된 클라이언트 연산 프로세스에 의해 액세스될 수 있다. 중요하기로는, 클라이언트 연산 프로세스는 사용할 프로토콜을 추정할 때 시행착오 없이 및 사용할 프로토콜에 대한 사전 구성된 이해 없이 적절한 통신 프로토콜을 선택할 수 있다. 보다 구체적으로, 도 4는 도 1의 시스템에서 서버 지정형 프로토콜 스택에 기반하여 다른 클라이언트 측 프로토콜 요소의 특정 프로토콜 스택을 선택하고 구성하는 프로세스를 도시하는 흐름도이다.

<28> 블록(410)에서 처음에, 바람직한 프로토콜 스택 구성은 서버 연산 프로세스로부터 검색될 수 있다. 이 점에 있어서, 클라이언트 연산 프로세스는 예컨대, 기본 디렉토리 인터페이스를 통해 바람직한 프로토콜 스택에 대한 인터페이스를 액세스함으로써 서버 연산 프로세스에 바람직한 프로토콜 스택 구성에 대해 질의할 수 있다. 결정 블록(420)에서, 바람직한 프로토콜 스택이 클라이언트 연산 프로세스 내에 이미 존재하는 프로토콜 스택 인스턴스와 유사한지를 판단할 수 있다. 만일 유사하지 않다면, 적절한 종속성 순서로 구성 내에서 지정된 각 요소를 갖는 새로운 프로토콜 스택 인스턴스가 생성될 수 있다. 또한, 바람직한 프로토콜 스택에 설명된 것처럼 요소의 속성이 설정될 수 있고, 요소의 특성에 대한 특성값도 바람직한 프로토콜 스택에 따라 더 설정될 수 있다.

<29> 그러나, 결정 블록(420)에서, 만일 바람직한 프로토콜 스택이 클라이언트 연산 프로세스 내에 이미 존재하는 프로토콜 스택 인스턴스와 유사하면, 블록(430)에서 기존의 프로토콜 스택 인스턴스가 바람직한 프로토콜 스택에 의해 지정된 속성과 동일한 속성을 포함하는 지를 판단할 수 있다. 만일 아니라면, 블록(450)에서 이전처럼 바람직한 통신 프로토콜 스택의 새로운 인스턴스가 클라이언트에서 생성될 수 있다. 그렇지 않으면, 블록(440)에서, 서버 연산 프로세스와의 통신을 위해 기존의 통신 프로토콜 스택 인스턴스가 재사용될 수 있다.

<30> 본 발명은 하드웨어, 소프트웨어 또는 하드웨어와 소프트웨어의 조합으로 구현될 수 있다. 본 발명의 방법 및 시스템의 구현은 하나의 컴퓨터 시스템에서의 집중 방식 또는 상이한 요소가 다수의 상호접속 컴퓨터 시스템을 통해 전개되는 분산 방식으로 구현될 수 있다. 어떤 종류의 컴퓨터 시스템 또는 본 명세서에 설명된 방법을 수행하기에 적합한 다른 장치는 본 명세서에 설명된 기능을 실행하기에 적합하다.

<31> 하드웨어 및 소프트웨어의 전형적인 조합은 로딩되고 실행될 때 본 명세서에 설명된 방법을 수행하도록 컴퓨터 시스템을 제어하는 컴퓨터 프로그램을 포함하는 범용 컴퓨터 시스템이 될 수 있다. 본 발명은 본 명세서에 설명된 방법의 구현을 가능하게 하는 모든 특성을 포함하며 컴퓨터 시스템에서 로딩될 때 이들 방법을 수행할 수 있는 컴퓨터 프로그램 제품에도 내장될 수 있다.

<32> 본 발명에서 컴퓨터 프로그램 또는 애플리케이션은 정보 처리 능력을 갖는 시스템이 a) 다른 언어, 코드 또는 기호로 변환, b) 다른 물질 형태로 재생성 중 하나 또는 둘 다를 실행한 즉시 또는 이후에 특정 기능을 수행하게 하는 인스트럭션 세트의 임의의 언어, 코드 또는 기호에 의한 임의의 표현을 의미한다. 중요한 것은, 본 발명은 다른 특정 형태에 구현됨에 있어서 상술한 명세 사항보다는 본 발명의 범주를 나타내는 후속하는 특허 청구 범위를 참조해야한다.

도면의 간단한 설명

<11> 도 1은 본 발명의 실시예에 따른 클라이언트를 액세스하는 서버 측 구성용으로 형성된 연산 시스템의 개략도이다.

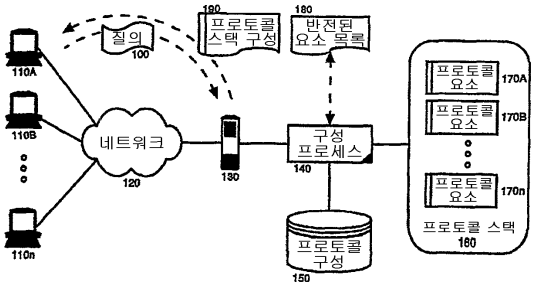
<12> 도 2는 도 1의 시스템에서 서버 측 구성 내의 서버 프로토콜 요소와 대응하는 클라이언트 프로토콜 요소를 연결(link)하는 프로세스를 도시하는 흐름도이다.

<13> 도 3은 도 1의 시스템에서 서버 내의 다른 프로토콜 요소의 프로토콜 스택(stack)을 초기화하는 프로세스를 도시하는 흐름도이다.

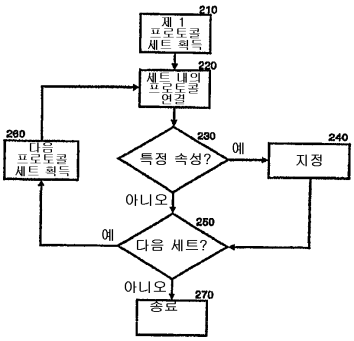
<14> 도 4는 도 1의 시스템에서 서버 지정형 프로토콜 스택에 기반하여 다른 클라이어트 측 프로토콜 요소의 특정 프로토콜 스택을 선택하고 구성하는 프로세스를 도시하는 흐름도이다.

도면

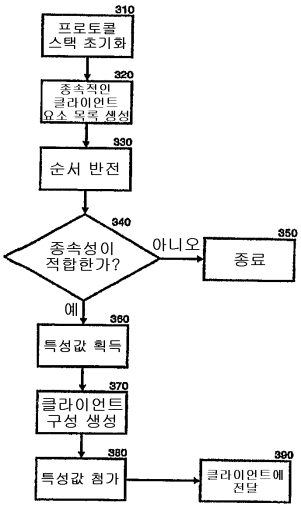
도면1



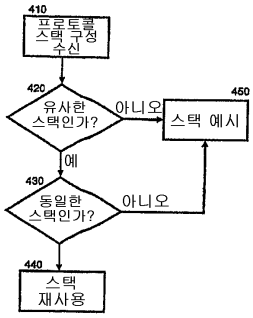
도면2



도면3



도면4


410
프로토콜 스택 구성 수신
420
유사한 스택인가?
아니오
450
스택 예시
예
430
동일한 스택인가?
아니오
440
스택 재사용