



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0106550
(43) 공개일자 2008년12월08일

(51) Int. Cl.

G06F 9/44 (2006.01) G06F 9/00 (2006.01)

(21) 출원번호 10-2008-7022940

(22) 출원일자 2008년09월19일

심사청구일자 없음

번역문제출일자 2008년09월19일

(86) 국제출원번호 PCT/US2007/004194

국제출원일자 2007년02월15일

(87) 국제공개번호 WO 2007/108879

국제공개일자 2007년09월27일

(30) 우선권주장

11/385,124 2006년03월21일 미국(US)

(71) 출원인

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이

(72) 발명자

시만, 조첸

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이

존스, 개리스 에이.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이

(뒷면에 계속)

(74) 대리인

양영준, 백만기

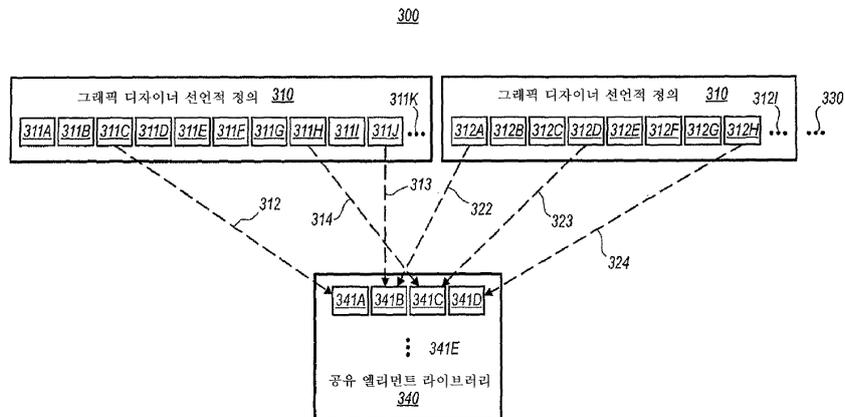
전체 청구항 수 : 총 20 항

(54) 그래픽 디자이너 재사용을 가능하게 하는 선언적 정의를 이용하기 위한 컴퓨터 프로그램 제품, 및 컴퓨팅 시스템, 컴퓨터 판독 가능 매체

(57) 요약

복수의 그래픽 모델링 툴들 사이에서의 그래픽 모델링 툴들의 하나 이상의 공유 컴포넌트들의 재사용이 개시된다. 그래픽 디자이너들, 즉 그래픽 모델링 툴들은 개체들, 개체 속성들, 및 개체 상호관계들이 시각화될 수 있는 모델들의 그래픽 표현들의 구성을 허용한다. 따라서, 각종 그래픽 모델링 툴들은 다른 그래픽 모델링 툴들의 부분들로 지원되고 채워질 수 있다.

대표도



(72) 발명자

켄트, 스투어트 제이. 에이치.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소
프트 웨이

쿠크, 스티븐 제이.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소
프트 웨이

특허청구의 범위

청구항 1

컴퓨터 실행 가능 명령어들을 갖는 하나 이상의 컴퓨터 판독 가능 매체를 포함하는 컴퓨터 프로그램 제품에 있어서, 상기 컴퓨터 실행 가능 명령어들은, 컴퓨팅 시스템의 하나 이상의 프로세서들에 의해 실행될 때, 상기 컴퓨팅 시스템이,

적어도 하나의 공유 엘리먼트의 선언을 포함하는 그래픽 디자이너의 선언적 정의를 번역하는 단계;

상기 그래픽 디자이너의 선언적 정의 외부에 있는 상기 공유 엘리먼트의 선언적 정의를 참조하는 단계; 및

상기 그래픽 디자이너의 선언적 정의 내의 상기 공유 엘리먼트의 선언을 이용하고 상기 공유 엘리먼트의 상기 외부에 있는 선언적 정의를 이용하여 상기 그래픽 디자이너 내에서 상기 공유 엘리먼트를 구현하는 단계

를 수행하게 하도록 구성된 것인 컴퓨터 프로그램 제품.

청구항 2

제1항에 있어서, 상기 하나 이상의 컴퓨터 판독 가능 매체는 물리적 메모리 매체인 컴퓨터 프로그램 제품.

청구항 3

제1항에 있어서, 상기 하나 이상의 컴퓨터 판독 가능 매체는 물리적 저장 매체인 컴퓨터 프로그램 제품.

청구항 4

제1항의 상기 컴퓨터 프로그램 제품을 포함하는 컴퓨팅 시스템.

청구항 5

제1항에 있어서, 상기 컴퓨터 판독 가능 매체는, 상기 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 컴퓨팅 시스템이,

상기 그래픽 디자이너의 상기 선언적 정의 내에 있는 비공유 엘리먼트의 선언적 정의를 번역하는 단계

를 더 수행하게 하는 컴퓨터 실행 가능 명령어들을 더 갖는 컴퓨터 프로그램 제품.

청구항 6

제5항에 있어서, 상기 컴퓨터 판독 가능 매체는, 상기 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 컴퓨팅 시스템이,

비공유 개체형의 선언적 정의를 이용하여 상기 비공유 엘리먼트를 구현하는 단계

를 더 수행하게 하는 컴퓨터 실행 가능 명령어들을 더 갖는 컴퓨터 프로그램 제품.

청구항 7

제1항에 있어서, 상기 공유 엘리먼트는 제1 엘리먼트 클래스의 제1 공유 엘리먼트이며, 상기 컴퓨터 판독 가능 매체는, 상기 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 컴퓨팅 시스템이,

상기 그래픽 디자이너의 상기 선언적 정의 내에서 제2 엘리먼트 클래스의 제2 공유 엘리먼트의 선언을 검출하는 단계;

상기 그래픽 디자이너의 상기 선언적 정의 외부에 있는 상기 제2 공유 엘리먼트의 선언적 정의를 참조하는 단계; 및

상기 그래픽 디자이너의 상기 선언적 정의 내의 상기 제2 공유 엘리먼트의 선언을 이용하고 상기 제2 공유 엘리먼트의 상기 외부 선언적 정의를 이용하여 상기 그래픽 디자이너 내에서 상기 제2 공유된 엘리먼트를 구현하는 단계

를 더 수행하게 하는 컴퓨터 실행 가능 명령어들을 더 갖는 컴퓨터 프로그램 제품.

청구항 8

제1항에 있어서, 상기 공유 엘리먼트는 제1 공유 엘리먼트이고 상기 그래픽 디자이너는 제1 그래픽 디자이너이며, 상기 컴퓨터 판독 가능 매체는, 상기 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 컴퓨팅 시스템이,

적어도 제2 공유 엘리먼트의 선언을 포함하는 제2 그래픽 디자이너의 선언적 정의를 번역하는 단계;

상기 제2 그래픽 디자이너의 선언적 정의 외부에 있는 상기 제2 공유 엘리먼트의 선언적 정의를 참조하는 단계; 및

상기 제2 그래픽 디자이너의 상기 선언적 정의 내의 상기 제2 공유 엘리먼트의 선언을 이용하고 상기 제2 공유된 엘리먼트의 상기 외부 선언적 정의를 이용하여 상기 제2 공유 엘리먼트를 구현하는 단계

를 더 수행하게 하는 컴퓨터 실행 가능 명령어들을 더 갖는 컴퓨터 프로그램 제품.

청구항 9

제1항에 있어서, 상기 제1 및 제2 공유 엘리먼트의 상기 선언적 정의들은 공유 라이브러리 데이터 구조로부터 액세스되는 컴퓨터 프로그램 제품.

청구항 10

제1항에 있어서, 상기 공유 엘리먼트는 제1 공유 엘리먼트이고 상기 그래픽 디자이너는 제1 그래픽 디자이너이며, 상기 컴퓨터 판독 가능 매체는, 상기 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 컴퓨팅 시스템이,

적어도 제2 공유 엘리먼트의 선언적 정의를 포함하는 제2 그래픽 디자이너의 선언적 정의를 번역하는 단계; 및

상기 제2 공유 엘리먼트의 상기 선언적 정의를 이용하여 상기 제2 공유 엘리먼트를 구현하는 단계

를 더 수행하게 하는 컴퓨터 실행 가능 명령어들을 더 갖는 컴퓨터 프로그램 제품.

청구항 11

제1항에 있어서, 상기 제1 및 제2 공유 엘리먼트의 상기 선언적 정의들은 상기 제2 그래픽 디자이너 내에 있는 공유 라이브러리 데이터 구조로부터 액세스되는 컴퓨터 프로그램 제품.

청구항 12

제1항에 있어서, 상기 공유 엘리먼트는 도메인 클래스, 도메인 클래스들 간의 관계, 및 열거 엘리먼트 중 적어도 하나를 포함하는 도메인 모델의 컴포넌트를 포함하는 컴퓨터 프로그램 제품.

청구항 13

제1항에 있어서, 상기 공유 엘리먼트는 형상, 커넥터, 또는 데코레이터 엘리먼트 중 적어도 하나를 포함하는 그래픽 주석을 포함하는 컴퓨터 프로그램 제품.

청구항 14

제1항에 있어서, 상기 그래픽 디자이너의 상기 선언적 정의는 맵핑 정의의 선언적 정의를 포함하는 컴퓨터 프로그램 제품.

청구항 15

컴퓨팅 시스템에 있어서,

하나 이상의 컴퓨터 판독 가능 매체;

하나 이상의 프로세서들을 포함하며,

상기 컴퓨터 판독 가능 매체는, 상기 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 컴퓨팅 시스템이, 그래픽 모델링 툴의 선언적 정의를 액세스하는 단계 - 상기 선언적 정의는 상기 그래픽 모델링 툴에 대한 복수의 엘리먼트들을 정의하며, 상기 복수의 엘리먼트들 중 적어도 하나는 선언적으로 정의된 개체 클래스들의 공유 라이브러리 내에 정의된 엘리먼트를 참조하여 정의됨 - ; 및 공유 엘리먼트의 선언적 정의의 번역에 기초하여 공유 개체 유형으로서 상기 복수의 엘리먼트들 중 적어도 하나를 식별하는 단계를 수행하게 하도록 구성된 컴퓨터 실행 가능 명령어들을 갖는 컴퓨팅 시스템.

청구항 16

데이터 구조를 갖는 하나 이상의 컴퓨터 판독 가능 매체에 있어서, 상기 데이터 구조는, 적어도 하나의 공유 엘리먼트의 선언을 포함하는 그래픽 디자이너의 선언적 정의 - 상기 공유 엘리먼트는 개체 클래스, 개체들간의 관계 유형, 및 개체 형태 중 적어도 하나를 포함함 - ; 상기 그래픽 디자이너의 선언적 정의의 외부에 있는 상기 공유 엘리먼트의 선언적 정의; 및 상기 그래픽 디자이너의 선언적 정의 내의 상기 공유 엘리먼트의 선언을 이용하고, 상기 공유 엘리먼트의 상기 외부 선언적 정의를 이용하여 상기 공유 엘리먼트를 구현하도록 구성된 번역 컴포넌트를 포함하는 하나 이상의 컴퓨터 판독 가능 매체.

청구항 17

제16항에 있어서, 상기 하나 이상의 컴퓨터 판독 가능 매체는 물리 메모리 및/또는 저장 매체인 하나 이상의 컴퓨터 판독 가능 매체.

청구항 18

제16항에 있어서, 상기 데이터 구조는, 상기 그래픽 디자이너의 선언적 정의 내에 있는 비공유 엘리먼트의 선언적 정의를 더 포함하는 하나 이상의 컴퓨터 판독 가능 매체.

청구항 19

제16항에 있어서, 상기 공유 엘리먼트는 제1 공유 엘리먼트이고, 상기 데이터 구조는, 상기 그래픽 디자이너 도메인 모델의 상기 선언적 정의 내의 제2 엘리먼트 클래스의 제2 공유 엘리먼트의 선언을 더 포함하는 하나 이상의 컴퓨터 판독 가능 매체.

청구항 20

제16항에 있어서, 상기 데이터 구조는, 상기 공유 엘리먼트 및 복수의 그래픽 디자이너들 사이에서 공유되는 복수의 다른 공유 엘리먼트들을 포함하는 공통 개체 데이터 구조를 더 포함하는 하나 이상의 컴퓨터 판독 가능 매체.

명세서

배경 기술

<1> 컴퓨팅 기술은 인류의 진보에 크게 기여해왔다. 컴퓨팅 시스템들은 데스크탑 컴퓨터, 랩탑 컴퓨터, PDA, 전화기, 및 예를 들어 냉장고 및 자동차와 같이 종래에 컴퓨팅 시스템인 것으로 간주되지 않았던 장치들도 포함하는 매우 다양한 물리적 형태들로 나타난다. 따라서, 컴퓨팅 시스템은 임의의 유형의 메모리 및 프로세서를 갖는

입의의 장치 또는 시스템일 수 있다.

- <2> 컴퓨팅 기술이 지원하는 공통 애플리케이션 중 하나는 그래픽 모델링 툴들("그래픽 디자이너들"로도 호칭됨)이다. 그래픽 모델링 툴들은 그래픽 모델들의 규칙적이고 효율적인 구성을 용이하게 한다. 그래픽 모델들은 상호관련 개체들의 모음(a collection of interrelated objects)에 대한 시각적 표현을 포함한다. 그래픽 모델들은 매우 다양한 도메인들에 의해 상이한 방법으로 이용될 수 있다. 예를 들어, 작업흐름들(workflows), 조직차트들, 전기 회로들, 소프트웨어 아키텍처들, 소프트웨어 흐름도들(software flowcharts)은 각각 그래픽 모델들을 이용하여 표현될 수 있다. 그래픽 모델링 툴들이 이용될 수 있는 문자 그대로 수천의 상이한 애플리케이션들이 있을 수 있다. 각각의 경우에, 개체들의 종류들 및 상호관계들은 매우 상이할 수 있다. 단일 도메인의 애플리케이션 내에서조차, 개체들 및 상호관계들이 디스플레이되는 방법에 있어서 명확한 선호들(definite preferences)이 있을 수 있다. 예를 들어, 한 은행은 경제 거래를 나타내는 어떤 그래픽 모델링 스타일을 선호할 수 있는 한편, 다른 은행은 경제 거래를 위한 다른 그래픽 모델링 스타일을 나타낼 수 있다.
- <3> 그래픽 모델링 툴들을 구축(building)하는 것은 어려우며 시간 소모적인 태스크이다. 전형적인 그래픽 디자이너의 구축은, 소프트웨어 개발자가 그래픽 주석(graphical notation)을 구현하는 디자인 표면(design surface), 사용자가 디자인 표면에 엘리먼트를 드래그 앤 드롭하는 것을 허용하는 툴박스, 디자인 표면 상의 개체들의 속성들을 나타내기 위한 메커니즘 - 상기 속성들은 사용자가 모델링을 시도중인 문제에 대한 의미있는 정보를 나타냄 -, 및 모델 데이터 사이에서 네비게이션하는 익스플로러같은 다른 UI(User Interface) 엘리먼트들을 구현하는 것을 포함할 수 있다.
- <4> 상이한 그래픽 디자인들에 대하여 매우 많은 수의 도메인들 및 선호들을 고려할 때, 각각의 도메인 및 선호를 위한 그래픽 모델링 툴들을 구축하는 태스크는 쉽지 않을 것으로 보인다.

발명의 상세한 설명

- <5> 발명의 개요
- <6> 그래픽 디자이너들, 즉 그래픽 모델링 툴들은, 일부가 시각화되는 엘리먼트들을 포함하는 모델들의 그래픽 표현의 구성을 허용한다. 반드시 필요한 것은 아니지만, 본 발명의 실시예들은 복수의 그래픽 디자이너들 사이에서 하나 이상의 공유 엘리먼트들을 공유하여 주어진 그래픽 디자이너에서 엘리먼트들의 더 큰 선택(a greater selection of elements)이 이용되도록 허용하는 것과 관련된다. 한 위치는 엘리먼트의 선언적 정의(declarative definition)를 가질 수 있는 한편, 상기 엘리먼트는 특별히 공유 엘리먼트의 선언적 정의를 포함하지 않고서도 공유 엘리먼트의 선언적 정의를 참조함으로써 그래픽 디자이너들의 선언적 정의들에 포함될 수 있다.
- <7> 본 개요는 상세한 설명에서 후술되는 개념들의 선택을 단순화된 형태로 소개하기 위하여 제공된다. 본 개요는 청구된 본 발명의 주요 특징들 또는 본질적 특징들을 식별하기 위한 것은 아니며, 청구된 본 발명의 범위를 결정하는 데에 도움이 되기 위한 것도 아니다.

실시예

- <17> 본 발명의 실시예들은 복수의 그래픽 모델링 툴들 사이에서의 그래픽 모델링 툴들의 선언적으로 정의된 엘리먼트들의 재사용(reuse)까지 확장된다. 먼저, 도 1에 관하여 본 발명의 원리들이 이용될 수 있는 예시적인 컴퓨팅 시스템이 설명될 것이다. 그 후, 후속하는 도면들에 관하여 본 발명의 원리들이 더욱 상세히 설명될 것이다.
- <18> 본 발명의 실시예들은 더욱 상세히 후술되는 바와 같이, 각종 컴퓨터 하드웨어를 포함하는 특수 목적 또는 범용 컴퓨터를 포함할 수 있다. 도 1은 본 발명의 특징들을 구현하기 위하여 이용될 수 있는 예시적인 컴퓨팅 시스템(100)의 개략도를 도시한다. 상기 설명된 컴퓨팅 시스템은 적합한 컴퓨팅 시스템의 단지 하나의 예일 뿐이며 본 발명의 기능 또는 이용 범위에 대한 어떠한 제한도 암시하려는 것이 아니다. 본 발명은 도 1에 도시된 컴포넌트들 중 임의의 하나 또는 조합에 관련있는 어떠한 종속성 또는 요건을 갖는 것으로도 해석되어서는 안된다.
- <19> 컴퓨팅 시스템들은 이제 점차적으로 매우 다양한 형태들을 하고 있다. 컴퓨팅 시스템들은, 예를 들어, 핸드헬드 장치들, 가전제품들, 랩탑 컴퓨터들, 데스크탑 컴퓨터들, 메인프레임들, 분산형 컴퓨팅 시스템들, 또는 심지어는 종래에 컴퓨팅 시스템인 것으로 간주되지 않았던 장치들일 수 있다. 본 설명부 및 청구범위에서, 용어 "컴퓨팅 시스템"은 적어도 하나의 프로세서, 및 프로세서에 의해 실행될 수 있는 컴퓨터 실행 가능 명령어들을

저장할 수 있는 메모리를 포함하는 임의의 장치 또는 시스템(또는 그들의 조합)을 포함하는 것으로 넓게 정의된다. 메모리는 어떠한 형태도 취할 수 있으며 컴퓨팅 시스템의 특징 및 형태에 의존할 수 있다. 컴퓨팅 시스템은 네트워크 환경을 통해 분산될 수 있고 복수의 구성 컴퓨팅 시스템들을 포함할 수 있다.

<20> 도 1을 참조하면, 가장 기본적인 구성에서, 컴퓨팅 시스템(100)은 적어도 하나의 처리 장치(102) 및 메모리(104)를 포함하는 것이 전형적이다. 메모리(104)는 휘발성, 비휘발성, 또는 그 둘의 일부 조합일 수 있는 시스템 메모리일 수 있다. 휘발성 메모리의 일례는 RAM(Random Access Memory)을 포함한다. 비휘발성 메모리의 예들은 ROM, 플래시 메모리 등을 포함한다. 용어 "메모리"는 본 명세서에서 물리적 저장 매체와 같은 비휘발성 대용량 저장 장치를 의미하기 위하여 이용될 수도 있다. 그러한 저장 장치는 휘발성이거나 비휘발성일 수 있으며, PCMCIA 카드, 자기 및 광 디스크, 자기 테이프 등을 포함할 수 있다(그러나 이에 제한되는 것은 아님).

<21> 본 명세서에 사용된 바와 같이, 용어 "모듈" 또는 "컴포넌트"는 컴퓨팅 시스템에서 실행하는 소프트웨어 개체들 또는 루틴들을 의미할 수 있다. 본 명세서에 기술된 다른 컴포넌트들, 모듈들, 엔진들, 및 서비스들은 컴퓨팅 시스템에서 실행하는 개체들 또는 처리들(예를 들면, 별도의 스레드들(threads))로서 구현될 수 있다. 본 명세서에 기술된 시스템 및 방법들은 소프트웨어로 구현될 수 있지만, 하드웨어, 및 소프트웨어와 하드웨어의 조합으로의 구현들도 가능하다.

<22> 이하의 설명에서, 본 발명의 실시예들은 하나 이상의 컴퓨팅 시스템들에 의해 수행되는 동작들에 관하여 기술된다. 그러한 동작들이 소프트웨어로 구현된다면, 그 동작을 수행하는 연관된 컴퓨팅 시스템의 하나 이상의 프로세서들은 컴퓨터 실행 가능 명령어들을 실행한 것에 응답하여 컴퓨팅 시스템의 동작을 지시한다. 그러한 동작의 일례는 데이터의 조작을 수반한다. 컴퓨터 실행 가능 명령어들(및 조작된 데이터)는 컴퓨팅 시스템(100)의 메모리(104)에 저장될 수 있다.

<23> 컴퓨팅 시스템(100)은 또한, 컴퓨팅 시스템(100)이 예를 들면, 네트워크(110)를 통하여 다른 컴퓨팅 시스템들과 통신하는 것을 허용하는 통신 채널들(108)을 포함할 수도 있다. 통신 채널들(108)은 통신 매체의 예들이다. 통신 매체는 통상적으로 반송파 또는 기타 전송 메커니즘과 같은 피변조 데이터 신호에 컴퓨터 판독 가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터 등을 구현하고 모든 정보 전달 매체를 포함한다. 예로서, 통신 매체는 유선 네트워크 또는 직접 배선 접속과 같은 유선 매체, 그리고 음향, 무선, 적외선, 및 기타 무선 매체와 같은 무선 매체를 포함하지만, 이에 제한되는 것은 아니다. 여기에서 사용된 컴퓨터 판독 가능 매체라는 용어는 저장 매체 및 통신 매체 둘 다를 포함한다.

<24> 본 발명의 범위 내의 실시예들은 또한 컴퓨터 실행 가능 명령어들 또는 데이터 구조들을 운반하거나 저장하기 위한 컴퓨터 판독 가능 매체를 포함한다. 그러한 컴퓨터 판독 가능 매체는 범용 또는 특수 목적 컴퓨터에 의해 액세스될 수 있는 임의의 이용 가능한 매체일 수 있다. 예로서, 그러한 컴퓨터 판독 가능 매체는 RAM, ROM, EEPROM, CD-ROM 또는 기타 광디스크 저장 장치, 자기 디스크 저장 장치 또는 기타 자기 디스크 저장 장치, 또는 범용 또는 특수 목적 컴퓨터에 의해 액세스되고 원하는 프로그램 코드 수단을 컴퓨터 실행 가능 명령어 또는 데이터 구조의 형태로 운반 또는 저장하기 위하여 이용될 수 있는 임의의 다른 매체와 같은 물리적 저장 장치 및/또는 메모리 매체를 포함할 수 있지만, 이에 제한되는 것은 아니다. 정보가 네트워크 또는 다른 통신 접속(배선 접속, 무선, 또는 배선 접속 또는 무선의 조합)을 통해 컴퓨터에 전송 또는 제공되는 경우, 컴퓨터는 그 접속을 적절히 컴퓨터 판독 가능 매체로 본다. 따라서, 임의의 그러한 접속은 적절히 컴퓨터 판독 가능 매체로서 호칭된다. 상기한 것들의 조합들도 컴퓨터 판독 가능 매체의 범위 내에 포함되어야 한다.

<25> 컴퓨터 실행 가능 명령어들은, 예를 들어, 범용 컴퓨터, 특수 목적 컴퓨터, 또는 특수 목적 처리 장치가 특정 함수(function) 또는 함수 그룹을 수행하게 하는 명령어들 및 데이터를 포함한다. 본 발명이 구조적 특징들 및/또는 방법적 동작들에 특정한 언어로 기술되었지만, 청구범위에 정의된 본 발명은 반드시 여기에 기술된 특징의 특징들 또는 동작들에만 제한되는 것은 아닌 것이 이해되어야 한다. 오히려, 여기에 기술된 특정한 특징들 및 동작들은 청구범위를 구현하는 예시적인 형태들로서 개시된다.

<26> 도 2는 그래픽 디자이너를 디스플레이하는 사용자 인터페이스(200)를 도시한다. 본 설명부 및 청구범위에서, "그래픽 디자이너" 또는 "그래픽 모델링 툴"은 하나 이상의 시각화된 개체들 및 상호관계들의 그래픽 모델의 구성을 허용하는 임의의 애플리케이션이다. 일부 경우들에서 단일 애플리케이션이 복수의 그래픽 디자이너들을 지원할 수 있다. 이전에 언급된 바와 같이, 상이한 애플리케이션 도메인들 및 선호들은 상이한 개체 클래스들, 상호관계들, 및 이들의 시각화를 보장할 수 있다. 본 발명의 원리들은 복수의 그래픽 디자이너들 사이에서 선언적으로 정의된 엘리먼트들의 공유를 허용한다. 따라서, 용어 "그래픽 디자이너" 및 "그래픽 모델링 툴"은 이제 도 2를 참조하여 설명될 예에 제한되는 것이 아닌 넓은 의미의 용어이다.

- <27> 사용자 인터페이스(200)는 그래픽 주석을 구현하는 디자인 표면(201), 사용자가 엘리먼트를 디자인 표면에 드래그 앤 드롭하도록 해주는 툴박스 페인(toolbox pane, 202), 디자인 표면에 개체의 속성들을 나타내기 위한 속성 페인(203), 및 모델 데이터 사이에서 네비게이션하기 위한 네비게이션 페인(204)을 포함한다.
- <28> 그래픽 디자이너(200)는 각종 모델들을 구성하는 데에 이용될 수 있다. 그러나, 각각의 그래픽 디자이너(200)는 소정 유형의 개체 클래스들, 관계들, 및 이들의 시각화를 허용할 것이다. 그래픽 디자이너는 선언적 정의를 이용하여 선언적 방식으로 정의된다. 그래픽 디자이너를 정의하기 위해 컴퓨터에 제공된 상세 명령어들의 리스트를 이용하는 대신, 그래픽 디자이너 선언적 정의는 각각 그래픽 디자이너의 연관된 엘리먼트를 정의하는 선언적 정의의 세트를 특징한다. "엘리먼트"는, 예를 들면, 개체 클래스들, 관계들, 이들의 시각화들, 또는 그래픽 디자이너의 임의의 다른 양태(시각화 또는 비시각화)를 포함하는 그래픽 디자이너의 임의의 양태일 수 있다. 선언적 정의의 목적은 그래픽 디자이너의 행위(behavior)를 기술하는 최소한의 효율적인 방법이 되어, 모든 부적절한 세부사항을 생략하는 것이다. 장래의 어느 시점에, 선언적 정의는 선언적 정의들에 의해 정의된 명세(specification)를 지키는 명령어 시퀀스들을 생성하기 위해 번역기 또는 코드 생성기에 의해 번역될 수 있다. 그래픽 디자이너의 선언적 정의를 나타내는 한 가지 방법은 XML(eXtensible Markup Language) 문서를 이용하는 것이다. 그러나, 선언적 정의는 바이너리 또는 다른 비텍스트(non-text) 형태들로도 표현될 수 있다. 그래픽 디자이너의 주어진 엘리먼트의 구현은 그래픽 디자이너의 세부사항들을 구현하는 실행 가능 코드를 변경시키는 대신, 단지 엘리먼트를 기술하는 선언적 정의를 변경시킴으로써 변경될 수 있다.
- <29> 방금 언급된 바와 같이, 그래픽 디자이너의 선언적 정의는 XML 문서의 형태일 수 있다. 일 실시예에서, 그래픽 디자이너의 선언적 정의는 특별한 문제 도메인(particular problem domain)에 특정되어 특정 개체들, 그들의 속성들, 및 개체들 간의 관계 유형들을 정의하는 개체 클래스들 형태의 도메인 모델 컴포넌트들을 포함할 수 있다. 문제 도메인에 의존하여, 이 엘리먼트들은 크게 변화할 수 있다. 예를 들어, 그래픽 디자이너가 금융 애플리케이션들에서 이용된다면, 개체 클래스들은, 통화, 지불, 송장 등을 포함할 수 있을 것이다. 그래픽 디자이너가 스케줄링 애플리케이션에 이용된다면, 개체 클래스는 태스크들, 완료 순서 의존도들(order of completion dependencies) 등을 포함할 수 있을 것이다. 사용자가 컴퓨터 네트워크의 구조를 모델링하기 위하여 디자이너를 구축한다면, 사용자는 "운영체제(Operating System)" 및 "저장 용량(Storage Capacity)"과 같은 속성들 및 "네트워크 연결(Network Connection)"과 같은 관계들을 이용하여 "서버(Server)"로 호칭되는 개체를 정의할 수 있다.
- <30> 도메인 모델들은 세 종류의 엘리먼트들: 도메인 클래스들, 도메인 클래스들 간의 관계들(자신들도 도메인 클래스들의 속성들을 가짐), 및 열거들(enumerations)을 포함할 수 있다. 이 엘리먼트들의 각각은 스키마 정의에서 나타날 수 있으며 다른 디자이너로부터의 엘리먼트를 참조하여 독립적으로 재사용될 수 있다.
- <31> 그래픽 디자이너 선언적 정의는 또한 디자인 모델의 개체들 및 관계들을 시각화하기 위하여 이용될 수 있는 주석 엘리먼트들(notation elements)을 포함할 수도 있다. 예를 들어, 컴퓨터 네트워킹 도메인에서, 그래픽 디자이너는 "서버" 개체를 시각화하는 아이콘을 이용할 수 있고 아이콘 옆에 텍스트로서 "서버" 개체의 명칭을 보인다. 네트워크 연결은 다른 "서버" 개체로의 화살표를 이용하여 시각화될 수 있다.
- <32> 주석 정의는 세 종류의 엘리먼트들: 형태들, 커넥터들(connectors) 및 데코레이터들(decorators)을 지정할 수 있다. 예를 들어, 형태들은 기하학적 형태들("사각형"과 같은 미리 정의된 도형들 이용), 개체가 이미지로서 표현되는 이미지 형태들, 또는 풍부한 행위(rich behavior)를 갖는 복잡한 형태일 수 있다. 커넥터들은 두 형태들을 연결하는 엘리먼트들이다. 커넥터들의 경우, 디자이너 구축자는 이 커넥터에 의해 어느 유형의 형태가 다른 어느 유형의 형태에 연결될 수 있는 것인지를 특정할 수 있다. 데코레이터들은 형태들 및 커넥터들에 텍스트 또는 아이콘들을 부가한다. 이 엘리먼트들의 각각은 또한 그래픽 디자인의 선언적 정의를 위한 스키마 정의에 나타날 수 있다.
- <33> 선언적 정의는 또한 맵핑 정의를 포함할 수 있다. 맵핑 정의는 도메인 모델의 어느 엘리먼트들이 주석의 어느 엘리먼트들과 연관되어 있는 것인지 특징한다. 예를 들어, 상기 언급된 샘플에서, 그래픽 디자이너는 "서버"에 대한 도메인 개체를 "서버" 개체를 시각화하기 위해 이용되는 형태로 맵핑할 수 있다.
- <34> 맵핑 정의는 네 종류의 엘리먼트들: TextDecoratorMaps, IconDecoratorMaps, ShapeMaps, 및 ConnectorMaps을 특징한다. TextDecoratorMaps 또는 IconDecoratorMaps는 데코레이터에 디스플레이된 값들을 개체의 값들로 맵핑한다. ShapeMaps는 형태 자체를 하나 이상의 개체들로 맵핑한다. 이 맵핑은 형태의 유형에 의존한다. 각각의 ShapeMap은 이 형태에 대한 DecoratorMaps의 리스트를 포함할 수 있다. ConnectorMaps은 커넥터를 두 도메인 개체들 간의 관계로 맵핑한다. 각각의 ConnectorMap은 이 커넥터에 대한 DecoratorMaps의 리스트를 포함할

수 있다. 다시 한 번, 이 엘리먼트들의 각각은 그래픽 디자인 선언적 정의에 대한 스키마 정의 파일에 나타날 수 있다.

- <35> 도 3은 공유 엘리먼트들이 있는 복수의 그래픽 디자이너들의 선언적 정의들을 포함하는 소프트웨어 아키텍처 환경(300)을 개략적으로 도시한다. 특히, 본 예에서, 제1 그래픽 디자이너 선언적 정의(310)는 수평 생략 표시(horizontal ellipses, 311K)로 표현된 바와 같이 잠재적인 다른 것들 가운데 엘리먼트들(311A 내지 311J)을 포함한다. 제2 그래픽 디자이너 선언적 정의(320)는 수평 생략 표시(321I)로 표현된 바와 같이 잠재적인 다른 것들 가운데 엘리먼트들(321A 내지 321H)을 포함한다. 수평 생략 표시(330)에 의해 표현된 바와 같이 그래픽 디자이너들의 다른 선언적 정의들이 있을 수 있다.
- <36> 그래픽 디자이너 선언적 정의들은 그 구성 엘리먼트들의 일부 선언적 정의들을 포함한다. 예를 들어, 제1 그래픽 디자이너 선언적 정의(310)는 엘리먼트들(311A, 311B, 311D, 311E, 311F, 311G 및 311I)에 대한 선언적 정의들을 포함한다. 제2 그래픽 디자이너 선언적 정의(320)는 엘리먼트들(321B, 321C, 321E, 321F 및 321G)에 대한 선언적 정의들을 포함한다.
- <37> 그래픽 디자이너 선언적 정의들은 엘리먼트들에 대하여 표현된 정의를 포함하지 않고, 엘리먼트들 중 일부에 대한 선언들을 포함할 수도 있다. 예를 들어, 제1 그래픽 디자이너 선언적 정의(310)는 공유 엘리먼트(341A, 341C 및 341B)의 대응하는 선언적 정의에 대한 대응하는 참조들(312, 314 및 313) 형태로 엘리먼트들(311C, 311H 및 311J)에 대한 선언을 포함한다. 제2 그래픽 디자이너 선언적 정의(320)는 공유 엘리먼트(341B, 341C 및 341D)의 대응하는 선언적 정의에 대한 대응하는 참조들(322, 323 및 324) 형태로 엘리먼트들(321A, 321D 및 321H)에 대한 선언들을 포함한다.
- <38> 따라서 공유 엘리먼트들에 대한 선언적 정의들은 엘리먼트의 완전한 선언적 정의를 포함하지 않고 단지 공유 엘리먼트를 참조하는 것에 의하여 각종 상이한 그래픽 디자이너들에서 재사용될 수 있다. 일 실시예에서, 공유 엘리먼트 선언적 정의들은 공유 엘리먼트들의 라이브러리(340) 내에 포함될 수 있다. 예를 들어, 이 경우에, 라이브러리는 수직 생략 표시들(341E)로 표현된 바와 같이 잠재적인 다른 것들 가운데 대응하는 엘리먼트들(341A 내지 341D)의 선언적 정의들을 포함하는 것으로 도시된다. 라이브러리(340)는 모든 그래픽 디자이너 선언적 정의들의 외부에 있을 수 있다. 그러나, 라이브러리(340)는 또한 단일의 그래픽 디자이너 선언적 정의 내에 완전히 포함될 수 있거나, 심지어는 복수의 그래픽 디자이너 선언적 정의들을 통해 분산될 수 있다. 공유 엘리먼트들은, 예를 들어, 1) 도메인 클래스, 도메인 클래스들 간의 관계, 및 열거 엘리먼트 중 적어도 하나를 포함하는 도메인 모델의 컴포넌트, 2) 형태, 커넥터, 또는 테코레이터 엘리먼트 중 적어도 하나를 포함하는 그래픽 주석 및/또는 3) 맵핑 정의일 수 있다.
- <39> 도 4는 본 발명의 원리들의 일 실시예에 따른 그래픽 디자이너 선언적 정의에서의 공유 엘리먼트 재사용을 위한 방법(400)의 흐름도를 도시한다. 방법(400)은 각각의 그래픽 디자이너 선언적 정의에 대하여 반복될 수 있다. 방법(400)은 도 3에 도시된 예시적인 아키텍처(300)를 빈번히 참조하여 설명될 것이다.
- <40> 주어진 그래픽 디자이너("그래픽 모델링 툴"로도 호칭됨) 선언적 정의에 대하여, 방법(400)은 그래픽 디자이너의 선언적 정의에 액세스한다(동작(401)). 예를 들어, 도 3의 그래픽 디자이너 선언적 정의(310)가 액세스된다. 그 후, 그래픽 디자인의 선언적 정의가 번역된다(동작(402)).
- <41> 번역하면서, 번역 컴포넌트는 그래픽 디자이너의 복수의 엘리먼트들을 평가할 수 있다. 특히, 엘리먼트들 각각에 대한 선언적 정의가 평가된다. 엘리먼트를 평가하기 위한 흐름은 엘리먼트가 그래픽 디자이너 선언적 정의의 내부에 정의되어 있는지 여부(엘리먼트들(311A, 311B, 311D, 311E, 311F, 311G 및 311I)의 경우에서와 같음), 또는 엘리먼트가 그래픽 디자이너 선언적 정의의 외부에 정의되어 있는 공유 엘리먼트인지 여부(엘리먼트들(311C, 311H 및 311J)의 경우에서와 같음)에 따라 상이할 것이다(결정 블록(410)). 엘리먼트는 엘리먼트 선언 자체에 기초하여 공유 또는 비공유 엘리먼트로서 식별될 수 있다. 본 설명부 및 청구범위에서, 컴포넌트의 "선언"은 선언적 정의에 대한 참조로서 정의되며, 선언적 정의를 포함하지는 않는다.
- <42> 엘리먼트가 외부 공유 엘리먼트라면(결정 블록(410)에서 "공유(Shared)"), 엘리먼트의 선언이 액세스되는데(동작(411)), 상기 선언은 엘리먼트의 선언적 정의의 위치에 대한 참조를 포함한다. 그 후, 선언적 정의를 포함하는 위치가 액세스되며, 거기에서 선언 정의가 참조된다(동작(412)). 그 후, 공유 디자이너 엘리먼트에 대한 코드가 그래픽 디자이너 선언적 정의 외부에 있는 공유 엘리먼트의 선언적 정의를 이용하여 생성된다(동작(413)). 예를 들어, 상기 코드는 공유 엘리먼트가 시각적 엘리먼트인 경우 디스플레이 상에 렌더링되도록 하고/거나 그것이 시각적 엘리먼트인지 비시각적 엘리먼트인지 여부와 무관하게 메모리 내에서 인스턴스화되게 할 수 있다.

디자이너 엘리먼트에 대한 코드 생성은 선언적 정의를 이용하는 어떠한 방식으로도 달성될 수 있다.

- <43> 한편, 엘리먼트의 선언적 정의가 그래픽 디자이너 선언적 정의 내부에 있다면(결정 블럭(410)에서 "내부(Internal)"), 선언적 정의는 번역 컴포넌트를 이용하여 번역되고(동작(421)), 디자이너 엘리먼트에 대한 코드가 생성된다(동작(422)). 코드 실행시, 엘리먼트는 렌더링 및/또는 메모리 내에서 인스턴스화되어 그것의 정의된 함수를 수행하게 된다.
- <44> 본 발명의 원리들 및 일부의 특정한 예들을 설명하였으며, 이제 도 5 내지 8에 관하여 더욱 구체적인 예가 설명될 것이다. 제1 그래픽 디자이너에서, 선언적 정의는 기업 환경의 비즈니스 분석 애플리케이션에서 사용된다. 대응하는 그래픽 디자이너에 의해 구축된 결과적인 그래픽 모델이 도 5에 도시된다. 도 5의 사용자 인터페이스는 "Customer"와 같은 다수의 비즈니스 주체들(business entities), 그들의 속성들 및 그들의 서로 간의 관계들을 도시한다.
- <45> 기초 디자이너 스키마(underlying designer schema)는 상기 디자이너에서 시각화되어 있는 도메인 클래스 "Business Entity"를 정의한다. 디자이너 최종 사용자(desiner end user)는 이 디자이너로 "Customer"와 같은 개체들을 생성하고 있다. 여기에 도메인 클래스 "BusinessEntity"의 정의가 있다. 도 6은 그래픽 트리로서 그래픽 디자인 선언적 정의를 도시한다.
- <46> 다른 그래픽 디자이너에서, 비즈니스 분석자는 비즈니스 처리 및 어느 비즈니스 주체들이 특정 처리 단계에서 이용되는 것인지를 정의할 수 있다. 도 7은 제2 그래픽 디자이너에서 발생한 그래픽 모델을 도시하는 사용자 인터페이스를 예시한다.
- <47> 도 7의 사용자 인터페이스는 비즈니스 처리를 예시한다. 상기 처리는 '인벤토리 체크(Check Inventory)'로 레이블링된 단계로 시작한다. 인벤토리 체크의 다이어그램 도형은 이 처리 단계에 수반되는 'Order' 또는 'Customer'와 같은 다수의 비즈니스 주체들을 보여준다. 이 개체들은 다른 디자이너의 도메인 모드에서 정의되고 본 그래픽 디자이너에서 재사용된다.
- <48> 도 8은 다른 그래픽 디자이너들에 의한 사용을 위해 공유 엘리먼트들이 정의될 수 있는 선언적 정의의 일부분을 정의하는 엘리먼트(부모 노드 "DsLibrary"를 가짐)를 포함하는 그래픽 디자인 선언적 정의(부모 노드 "Ds1"을 갖는 것으로 예시됨)의 계층 구조의 일례를 도시한다.
- <49> 방법들은 도 4에 대하여 설명되었지만, 본 발명의 원리들은, 서버 컴퓨팅 시스템의 하나 이상의 프로세서들에 의해 실행되는 경우, 컴퓨팅 시스템이 이 방법들의 전부 또는 일부 또는 어느 하나 또는 양쪽을 수행하게 하는 하나 이상의 컴퓨터 실행 가능 명령어들을 저장하는 하나 이상의 컴퓨터 판독 가능 매체를 포함하는 컴퓨터 프로그램 제품에게까지 확장된다. 예를 들어, 도 1을 참조해보면, 메모리(104) 및 통신 채널들(108)이 이러한 컴퓨터 판독 가능 매체의 예들을 나타낼 수 있다. 메모리(104)는 물리적 저장 장치 및/또는 메모리 매체 형태의 물리적 컴퓨터 판독 가능 매체의 일례를 나타낸다. 본 발명의 원리들은 또한 도 4의 방법들의 모든 부분들을 수행하도록 하드웨어, 소프트웨어, 또는 하드웨어와 소프트웨어의 조합을 통해 구성되는 컴퓨팅 시스템들 자체까지 확장된다.
- <50> 따라서, 본 발명의 원리들은 복수의 그래픽 디자이너 선언적 정의들 사이에서의 공유 엘리먼트의 재사용을 허용한다. 본 발명은 그 취지 또는 본질적 특징들로부터 벗어나지 않고 다른 특정 형태들로 구현될 수 있다. 상기 설명된 실시예들은 모든 측면들에서 제한이 아닌 예로서만 간주된다. 따라서, 본 발명의 범위는 상기 설명부에 의해서가 아닌 첨부 청구범위에 의해 나타내어진다. 청구범위의 균등의 의미 및 범위 내에 있는 모든 변화들은 본 발명의 범위 내에 포함되는 것이다.

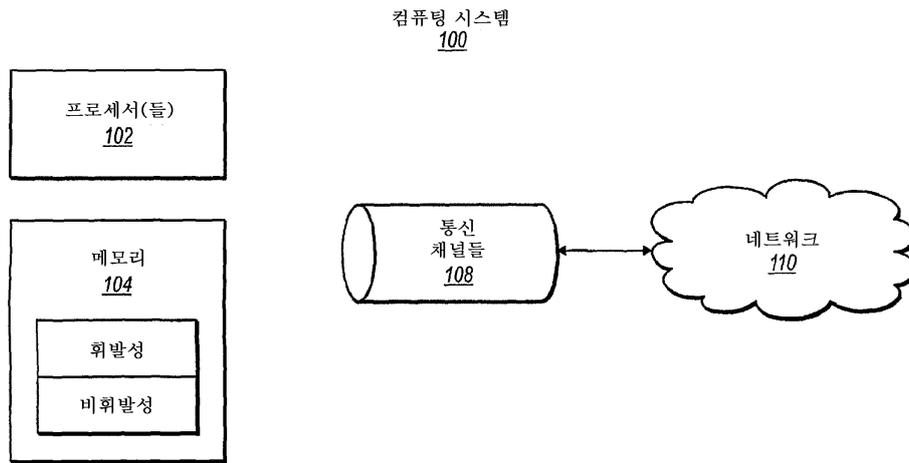
도면의 간단한 설명

- <8> 본 발명의 실시예들을 더욱 특별히 기술하기 위하여 첨부도면들이 이용된다. 이 도면들은 단지 본 발명의 전형적인 실시예들을 묘사하는 것이므로 본 발명의 범위를 제한하는 것으로 간주되지 않는 것을 이해하면서, 본 발명의 실시예들은 첨부도면들을 이용하여 부가적인 전문성을 가지고 상세하게 기술 및 설명될 것이다.
- <9> 도 1은 본 발명의 원리들의 실시예들이 동작할 수 있는 컴퓨팅 시스템을 도시하는 도면.
- <10> 도 2는 그래픽 디자이너를 디스플레이하는 사용자 인터페이스를 도시하는 도면.
- <11> 도 3은 그래픽 디자이너들의 엘리먼트들이 공유되는 복수의 그래픽 디자이너들의 선언적 정의들을 포함하는 환경을 도시하는 도면.

- <12> 도 4는 본 발명의 원리들의 일 실시예에 따라 공유 엘리먼트들을 갖는 그래픽 디자이너를 구현하기 위한 방법의 흐름도를 도시하는 도면.
- <13> 도 5는 그래픽 디자이너의 사용자 인터페이스를 도시하는 도면.
- <14> 도 6은 도 5의 사용자 인터페이스의 그래픽 트리를 도시하는 도면.
- <15> 도 7은 다른 그래픽 디자이너의 사용자 인터페이스를 도시하는 도면.
- <16> 도 8은 공통의 공유 엘리먼트 라이브러리의 계층 구조를 도시하는 도면.

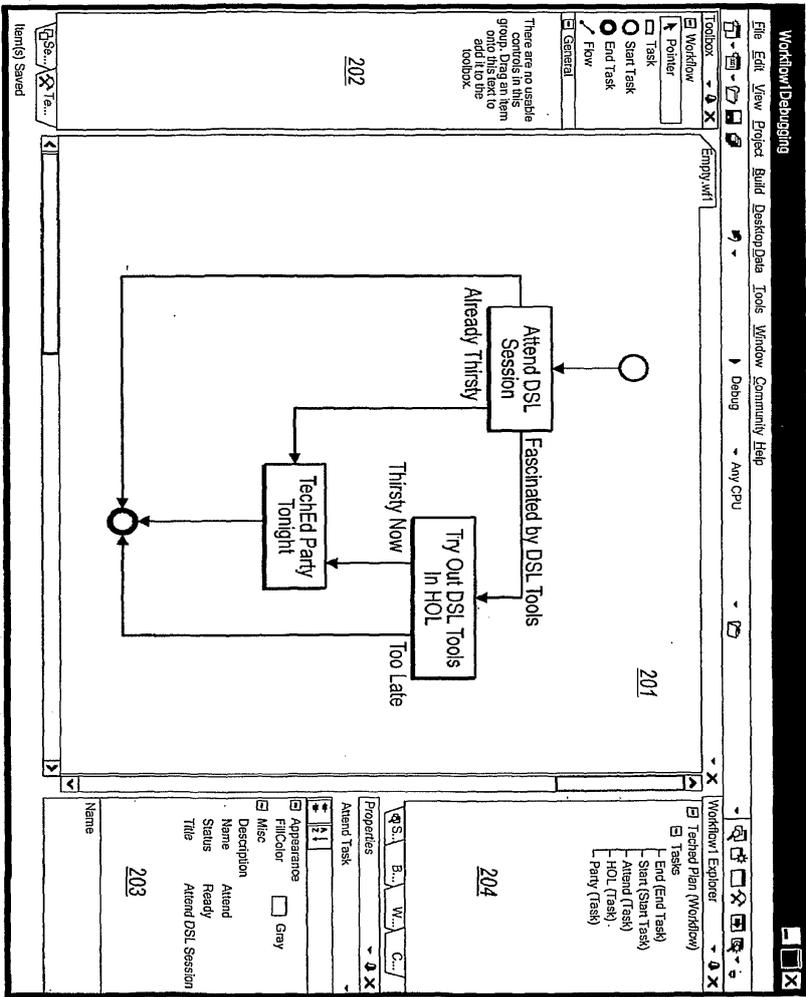
도면

도면1

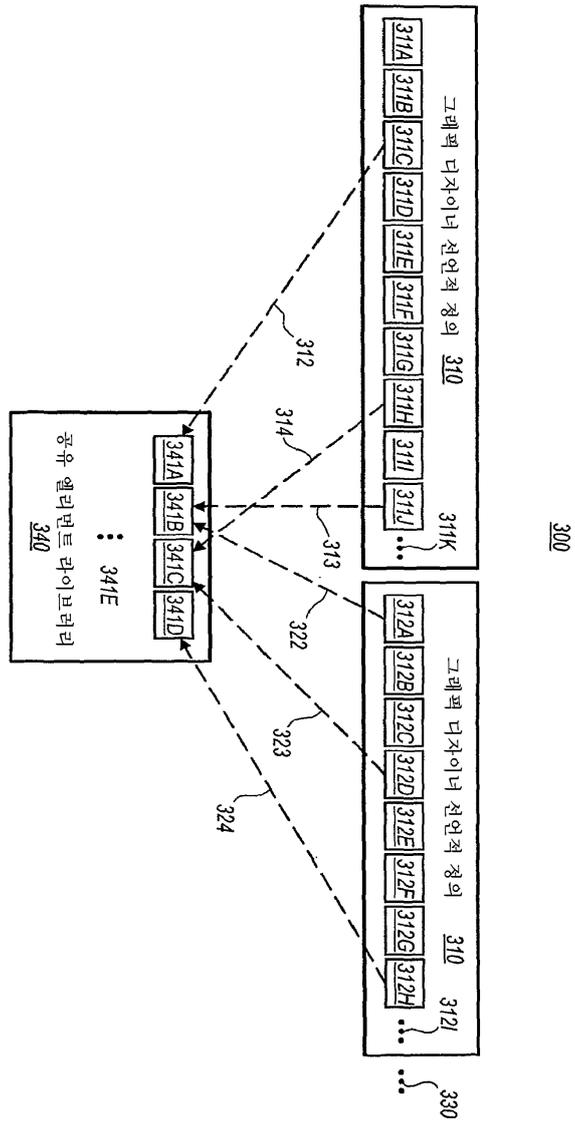


도면2

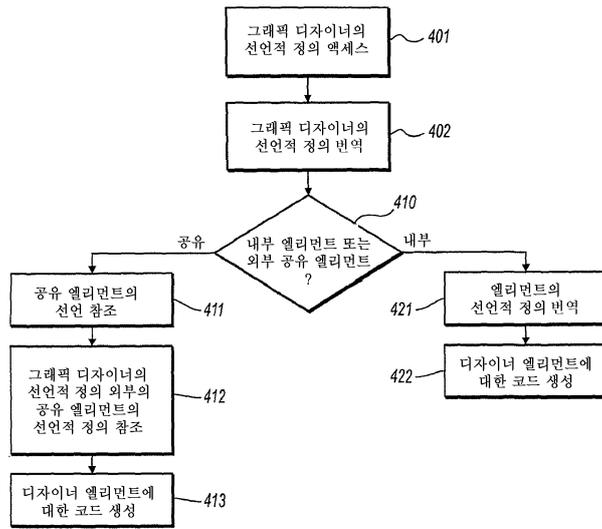
200



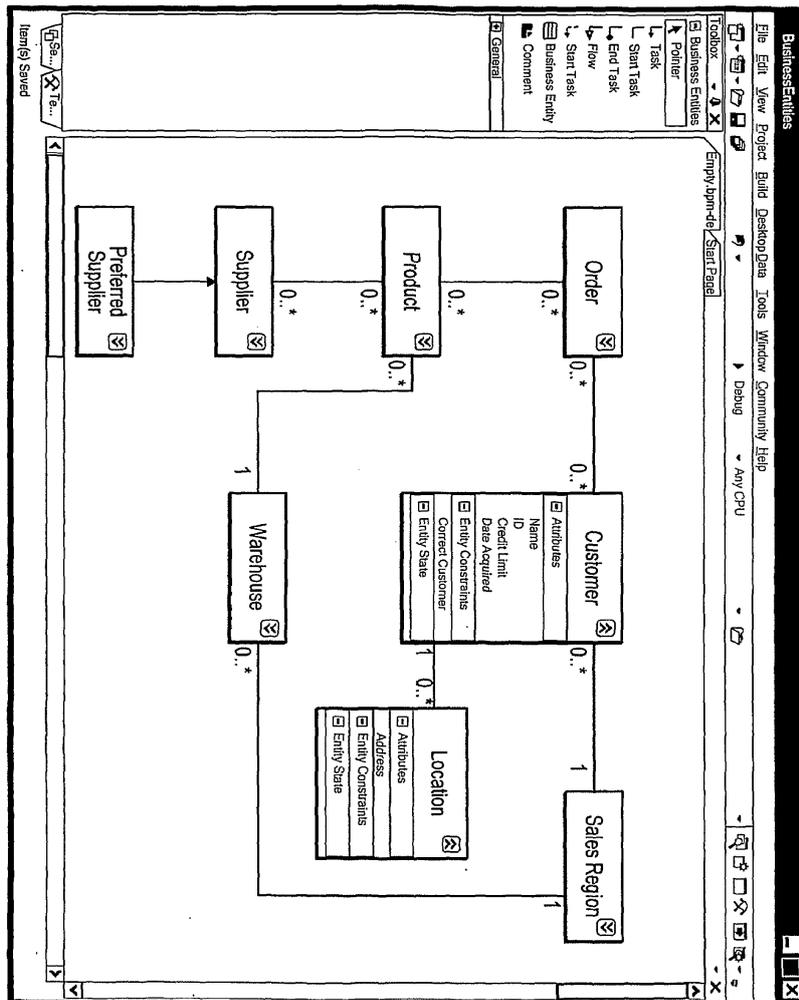
도면3



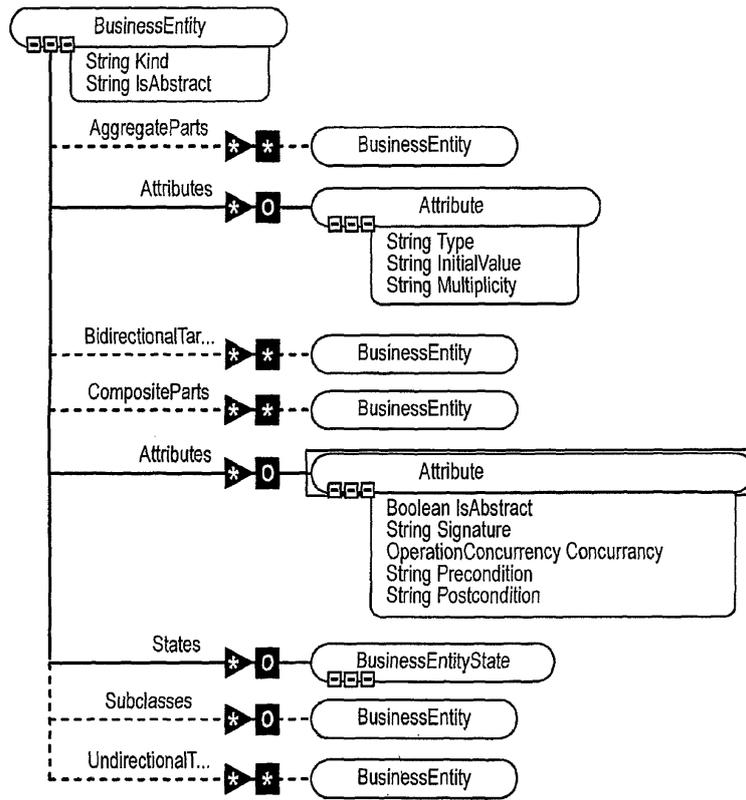
도면4



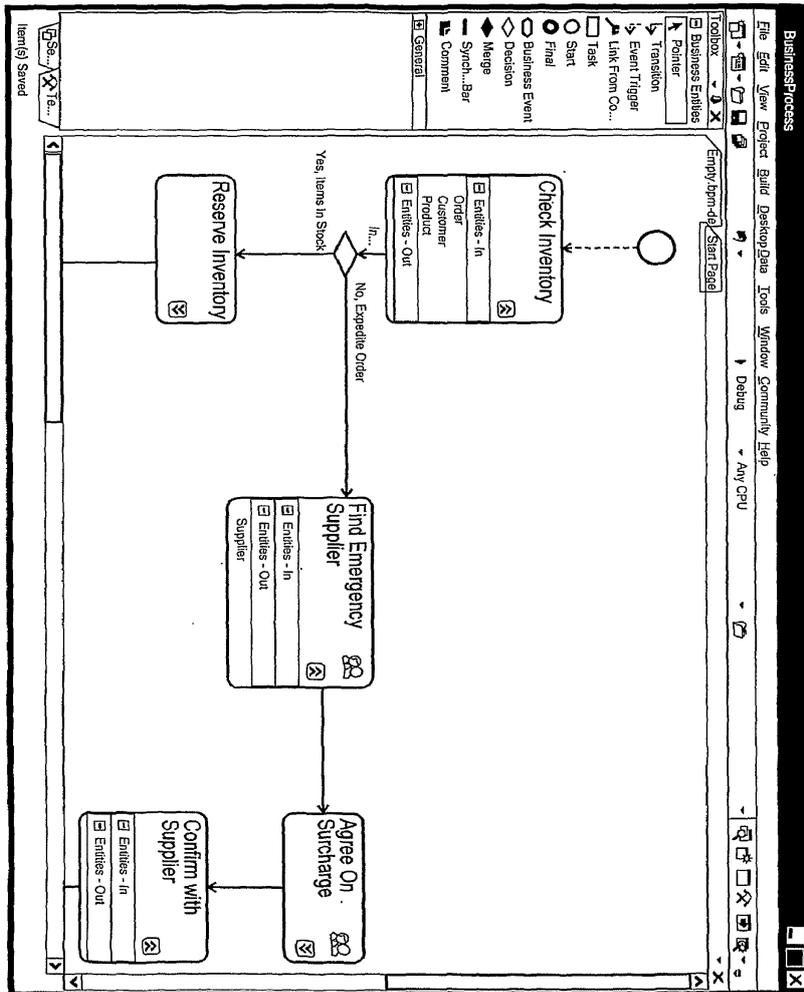
도면5



도면6



도면7



도면8

