

SPICE 참조모델 요구사항을 지원하는 데이터 모델링 기법에 관한 연구

정 규 장*

A Study on Data Modeling Techniques for Control Requirements of SPICE Reference Model

Kyu-Jang Chung *

요 약

객체 모델링 기법의 그래픽 표현을 이용하여 자료의 추상화, 캡슐화, 모듈화, 계층화 할 수 있는 새로운 그래픽 정보시스템 개발 기술이 절실히 필요하다. 그래픽 자료의 추상화 방법을 개선하기 위하여 복합 객체 기술로 자료의 추상화와 계층화 개념을 기반으로 모델링하였으며, 메쉬, 레이어, 세그먼트, 인스턴스 등과 같은 여러 가지 도형요소의 클래스들을 지원하는 분류화와 다중상속 관계모델을 제안한다. 객체 모델링 기법과 스페이스 참조 모델을 이용하여 간단한 그래픽 정보시스템 개발사례를 통하여 소프트웨어 개발 주기와 소프트웨어 유지보수 비용을 줄일 수 있는 요구사항을 지원하는 객체 표현 방법의 데이터 모델링 기법을 비교하고 평가한다.

Abstract

there needs a new Geographic information system development Technology of the abstraction, encapsulation, modulation and hierarchy using Graphic representation of object modeling Technique. The method is based on composite object of Graphic data with the hierarchy concepts and abstraction of Graphic information in order to improve data abstraction of the graphic data file and described concept of multiple inheritance and classification that supports a wide variety of graphic class such as mesh unit, layer, segment and so on. in simple case of software development using SPICE model and object modeling techniques. this thesis suggested object representation of Graphic data which can reduce software development life cycle and the cost of software maintenance.

▶ Keyword : OMT, SPICE reference model, requirement engineering, data modelling.

• 제1저자 : 정규장
• 접수일 : 2004.05.25, 심사완료일 : 2004.08.21
* 남도종합기술(주) 부사장, 경인교대 겸임교수

I. 서론

종래의 구조적 프로그래밍, 구조적설계, 구조적 분석은 문제범위를 함수적인 속성위주로 데이터 구조와 논리들로서 해결하였으나, 1990년 초반 Booch, martin and odell, Rumbaugh의 객체지향분석과 설계기법은 자료절차를 묶어 객체라는 개념을 사용하여 현실세계를 표현하고 모델링하는 기법으로 모든 소프트웨어 구성요소를 그래픽 표기법을 이용한 추상화, 캡슐화, 모듈화, 계층화 등의 일관된 객체지향 개념을 적용하여 분석, 설계 및 구현단계의 전과정에 사용된다. J. Rumbaugh의 객체 모델링 기법(OMT: Object Modeling Techniques)이 있다[1]. 본 연구의 목적은 서로 상이한 구조를 갖고, 자료의 추상화 표현이 제공되지 않는 지형정보시스템(GIS: Geographic Information System)의 분석 및 설계과정에 모델링기법과 스파이스(SPICE: Software Process Improvement and Capability Determination) 모델의 요구사항 관리를 접목하여 지형정보의 추상화 객체모델링과 지형정보의 계층구조를 표현하는데 있으며 개발과정에서 요구사항 변경을 효율적으로 관리할 수 있는 기법을 제안한다. 본 논문의 구성은 2절에서는 관련연구로 OMT와 GIS 및 스파이스 참조모델을, 3절에서는 그래픽정보의 추상화 표현, 4절에서는 그래픽정보의 계층구조, 5절에서는 결론을 기술하였다.

II. 관련연구

현실세계의 문제를 객체지향개념을 적용하여 분석, 설계, 구현할 수 있는 데이터모델링에 관한 객체모델링 기법, 외국의 데이터 모델, 스파이스 참조모델을 살펴본다.

2.1 객체 모델링 기법

구조적인 자료의 양상(aspect)을 나타내는 객체모델, 임시

적인 행위중심의 제어양상을 보여주는 동적모델, 변환 시켜 줄 함수의 양상을 나타내는 함수모델이 있다. 그러므로 기존의 정보모델, 구조화 분석과 구조적 설계 방법과 거의 흡사하다. 즉 객체모델은 기존의 E-R모델 및 ERD와 흡사하며, 동적모델과 기능모델은 상태전이 다이어그램(STD), 데이터 흐름 다이어그램(DFD)과 각각 대응된다.

객체모델링 기법은 객체모델을 가장 중요시하고, 동적모델과 기능모델은 객체모델을 상호 보완하는데 필요하다.

객체모델은 객체들과 객체들의 관계, 객체들을 표현하는 클래스들의 특성을 정의해주는 속성과 연산, 객체 인스턴스의 물리적인 개념적인 연결로 시스템의 정적구조를 표현해주는 모델이다. 객체모형도는 객체, 클래스, 다른 객체들과의 관계를 모델링하기 위해 형식화된 그래픽 표기법이다. 객체모형도는 추상적인 모델링뿐만 아니라 실제 프로그램을 설계하는데 유용하다. 객체모형도에는 클래스 모형도와 인스턴스 모형도로 분류된다. 클래스 모형도는 모든 가능한 인스턴스를 표현해 줄 수 있는 스키마, 패턴 같은 역할을 하며, 인스턴스 모형도는 문서화에 사용된다. 연산은 하나의 클래스로 표현되는 객체들에 적용되거나 혹은 객체에 의하여 사용되는 함수 또는 변환이며, 매소드는 클래스안에 있는 연산의 구현이므로 하나의 클래스를 정리하여 표현한다.

동적모델은 사건, 상태, 상태모형도로 객체들의 구조와 다른 객체들과의 관계를 표시해주며, 사건은 외부적인 자극을 나타내며 상태는 객체의 값들을 나타내준다. 함수 모델은 다수의 데이터 흐름 모형도(DFD)들로 구성되어 단지 계산 결과들을 명시하며, 객체모델의 연산들과 동적모델의 행위들 그리고 객체모델의 제약조건 등을 기술한다. 데이터 흐름 모형도는 프로세스들을 통해 소스에서 목적지 까지 데이터 값의 흐름을 나타내는 그래프이며, 제어에 관련된 정보는 동적 모델에서 표현한다. 구성요소들은 데이터 값을 변환 시켜주는 프로세서, 데이터의 이동을 표시해주는 데이터흐름, 데이터를 생성하고 소모하는 객체인 액터, 데이터를 저장하는 저장소등이 있다. 함수모델측면에서 객체모델은 함수모델내의 액터, 데이터저장소, 데이터흐름 등의 구조를, 동적모델은 프로세스들이 수행하는 순서를 보이는 것이다. 객체 모델 측면에서 함수모델은 객체 모델에 있는 클래스들의 연산들과 각 연산의 아구먼트들을 보임으로서 클래스들 간의 공급 관계를 보이고, 동적모델은 각 개체의 상태들과 상태의 변화가 있을 때 수행되는 연산을 보여준다. 동적모델 측면에서는 함수 모델은 동적모델에 정의 되어 있지 않은 하위 행위들과 행위들의 정의를 보여주고, 객체모델은 어느 것이 상태를 바꾸고 연산을 받는가를 보여주고 있다[2][3].

2.2 스파이스 모델

SPICE로 불리는 ISO/IEC TR 2 15504는 9개의 표준으로 구성되어 있으며, 소프트웨어의 프로세스에 대한 계획, 관리, 감시, 통제, 개선을 위한 능력 심사와 프로세스 개선을 목적으로 한다[4]. 그리고 프로세스 심사의 기준이 되는 참조 모델은 프로세스와 프로세스 능력의 2개 차원으로 구성되어 있다. 프로세스 차원은 소프트웨어 프로세스를 평가하기 위해 프로세스 범주를 고객-공급자 프로세스(CUS), 공학 프로세스(ENG), 지원 프로세스(SUP), 관리 프로세스(MAN), 조직 프로세스(ORG) 등 다섯 가지로 구분하고, 각 프로세스 범주는 다시 세부적인 프로세스들로 구분된다. 그리고 프로세스의 능력 차원은 Level 0 ~ Level 5의 여섯 단계 프로세스 능력 수준과 아홉 개의 PA(Process Attribute)로 구성되어 있다[5]. 세부 프로세스 중에서 요구 사항과 직접 관련되는 프로세스는 CUS.3(요구 사항 도출), ENG.1.1(시스템 요구 사항 분석 및 설계), ENG.1.2(소프트웨어 요구 사항 분석) 등이 있는데, ENG.1.1과 ENG.1.2는 공학 프로세스의 개발 공정에 속하며, 전통적 개발 방법론의 시스템 요구 분석과 소프트웨어 요구 분석 단계에 대응하여 주로 개발자의 관점을 대상으로 프로세스 평가를 위한 기본활동들을 정의한다. 그리고 CUS.3은 고객과 공급자의 관점을 기준으로 하기 때문에 전체 생명 주기와 연관되고 요구 사항의 도출과 분석 그리고 변경 관리가 주요 대상이 되는 프로세스이다 [6][7].

(그림 1)의 ENG.1.1 시스템 요구 사항 분석 및 설계의 기본 활동 순서에서 ENG.1.1(시스템 요구 사항 분석 및 설계 프로세스)은 시스템에 대한 요구 사항과 아키텍처를 정립하는 프로세스로, 시스템 요구 사항 식별, 시스템 요구 사항 분석, 시스템 아키텍처 기술, 요구 사

항의 할당, 공표 전략 수립, 시스템 요구 사항 전달, 추적성 수립 등의 7 가지 기본 활동으로 구성된다. 이 프로세스는 사용자의 요구 사항이 시스템의 어느 구성 요소에서 소프트웨어적으로 구현되는 지를 식별하게 하며, 시스템의 요구 사항은 전체 개발 프로세스에 관련되어 사용자의 요구 사항이 전해질 수 있도록 한다. 그리고 프로세스의 결과로 문제 영역에 대한 우선 순위가 결정되고, 그 결과가 승인되며 개발 과정에서 변형된다. ENG.1.1에서 BP1(시스템 요구 사항 식별)은 고객의 요구 사항을 기초로 하여 이를 시스템의 기능과 능력을 제시할 수 있는 정형화된 형태로 변환하고 문서화하여 시스템 명세서(규격서)를 도출하며, BP2(시스템 요구 사항의 분석)는 요구 사항 체계 규격서 상의 시스템 구성 요소와 그들 간의 인터페이스를 식별하여 우선 순위가 제기된 요구 사항을 분석한다. 그리고 BP3(시스템 아키텍처 기술)은 사용자가 접할 수 있는 계층 즉 최상위 수준의 아키텍처를 기술하고, BP4(요구 사항 할당)는 모든 시스템 요구 사항을 최상위 수준 시스템 구조의 구성 요소에 할당하며, BP1 ~ BP4의 수행 결과로 시스템 구조에서 각 구성 요소의 위치와 그 구성 요소와 관련된 요구 사항을 설명하는 문서화된 제품 형상을 생성한다. BP5(공표 전략의 발전)는 시스템 요구 사항의 우선 순위를 결정하고 개발하려는 구성품의 개발 완료 전략을 발전시킨다. 시스템 요구 사항을 장래의 시스템 개발과 관련시켜 개발 완료 전략을 수립한다. BP6 (시스템 요구 사항 전달)은 시스템 요구 사항과 그 요구 사항에 대한 변경을 모든 당사자에게 알리기 위한 의사 소통 체계를 수립하며, BP7(추적성 수립)은 고객의 요구와 시스템 요구 사항간의 추적성을 수립한다.

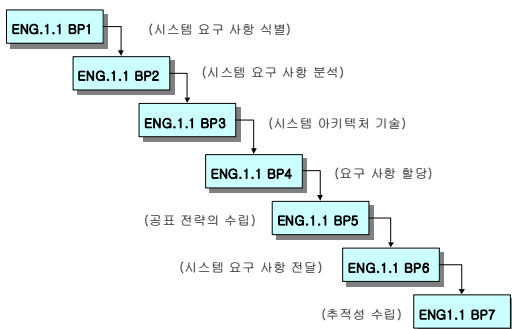
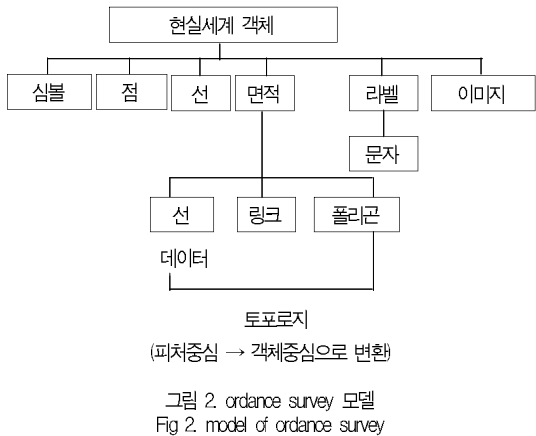


그림 1. 기본 활동 순서
FIG 1. fundamental procedure of activity

2.3 외국의 데이터 표준 모델

그래픽 정보시스템은 여러 가지의 지형정보와 속성정보를 데이터베이스화하여 수정, 저장, 검색하는 시스템으로 이용분야별 분류방법과 이용목적에 따라 지도 작성용 시스템, 지도의 도형정보를 기초로 하는 시스템, 지도의 도형위치를 기초로 한 시스템으로 크게 3가지 시스템 형태로 분류되고 있으며, 영국에서 표준으로 사용되고 있는 데이터 모델은 (그림 2)와 같다.



III. 그래픽 정보의 추상화 표현

시스템 개발의 모든 생명 주기에서 새로운 요구 사항이 생겨나고 기존의 요구 사항에 대한 변경이 발생할 수 있는데, 전체 50% 이상의 요구 사항이 실제 서비스 단계로 구현되기 전에 변경되는 경우가 빈번하게 발생한다[8]. 요구 사항의 변경은 요구 사항 자체의 에러나 요구 사항간의 상충, 요구 사항 처리 과정의 잘못된 이해, 설계나 구현상의 문제 등으로 인하여 발생하고, 새로운 요구 사항은 관련 당사자들이 개발 중인 시스템을 좀 더 이해하게 됨에 따라 나타날 수 있다. 대부분의 요구 사항 변경은 환경의 변화, 경쟁 제품의 등장 및 법률과 규정의 변경 등과 같은 외부 환경의 변화로 인하여 발생하는 경우가 많고, 요구 사항이 변경되면 시스템의 설계와 구현이 변경되어야 한다. 지형정보의 추상화는 벡터 그래픽의 입력과정에서부터 이루어지며 그 과정은 아래 (그림 3)과 같다.

개발 방법론의 요구 사항 관리 범위를 전체 개발 생명 주기로 확대하고 지속적인 요구 사항의 변경을 지원하기 위해서는 요구 공학적 기법을 개발 방법론에 적용하기 위한 기법이 필요하다. 이러한 기법은 요구 공학적 기법과 요구 사항 관리 도구의 개발 방법론 적용을 용이하게 하고, 나아가 개발 조직의 프로세스 능력과 성숙도 향상에 도움이 된다. 국내의 개발 방법론에 대한 접목은 한국전산원을 중심으로 한국정보통신기술협회와 한국전자통신연구원 등에서 이루어지고 있는데, 한국전산원에서는 국가 표준으로 적용

되는 '관리기법/1' 방법론과 이를 바탕으로 하는 감리 지침에 관한 연구를, 그리고 한국전자통신연구원에서는 국산 방법론인 '마르미'에 대한 개발로 필드에서 사용되고 있다. 이외에도 일부 프로젝트에서는 정보공학 방법론이 적용되고 있으며, 최근에는 객체 지향 개발 방법론과 UML(Unified Modeling Language)을 적용한 객체 지향 개발 사례가 늘어나고 있다.

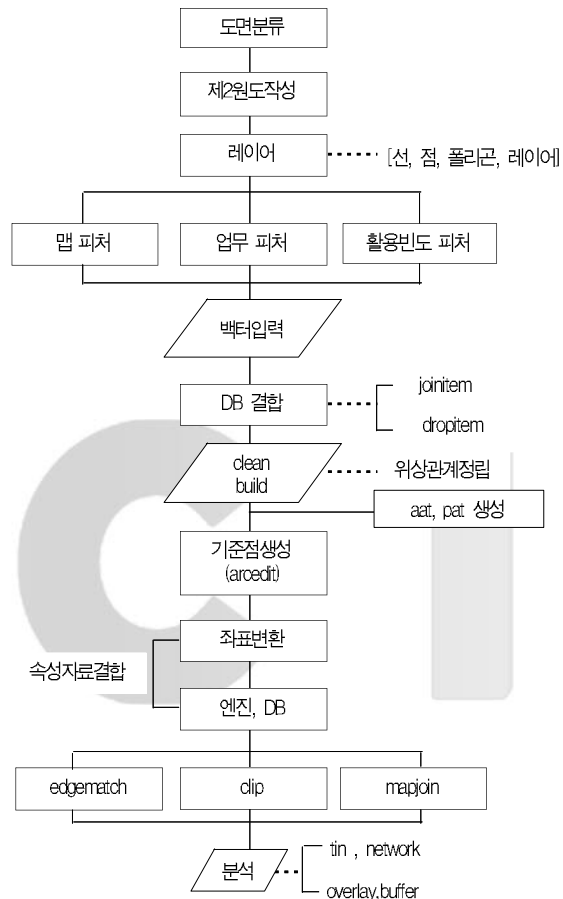


그림 3. 정보의 추상화 절차
FIG 3. abstraction procedure of information

해외에서는 국제적인 표준으로 사용되고 있는 ISO/IEC 12207 소프트웨어 생명 주기 공정 표준(software life cycle process standard)과 미 국방성의 정보 시스템 개발 표준으로 제정된 MIL-STD-498 등과 함께, 주요 컨설팅 회사에서 개발한 Method/1, 4Front, Fusion, SUMMIT 등의 방법론이 활용되고 있다[9].

세그먼트와 세그먼트 집합은 멀티미디어 시스템구축에서 가장 빈번히 사용되는 최소단위 메쉬에서 표현하는 정보이므로 계층구조와 추상화 과정이 중요한 요소가 된다. 도면의 기본요소는 아크, 점, 폴리곤에 클래스로 백터값으로 입력될 때 AAT와 PAT에 대한 메타 테이블이 만들어져 의미를 갖는 정보 단위로써 세그먼트형태로 구성되어진다. 이 논문에서는 세그먼트와 세그먼트 그룹 프레임용 (그림 4)와 같이 지형정보의 객체를 정의할 수 있어 추상화표현이 가능하게 할 수 있다[10].

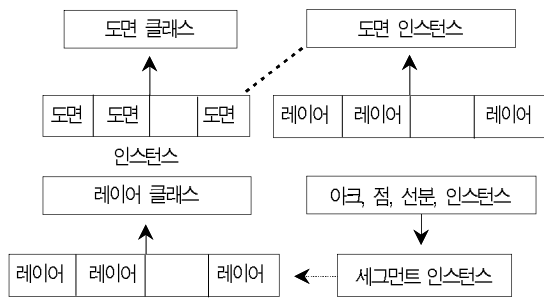


그림 4. 그래픽 정보의 추상화
FIG 4. abstraction of graphic information

도면, 메쉬 레이어, 세그먼트, 도형요소의 클래스들을 객체모델링기법의 분류화와 집합화(aggregation)의 개념을 이용하여 지형정보시스템의 정보를 계층구조로 표현하면 (그림 5)와 같다.

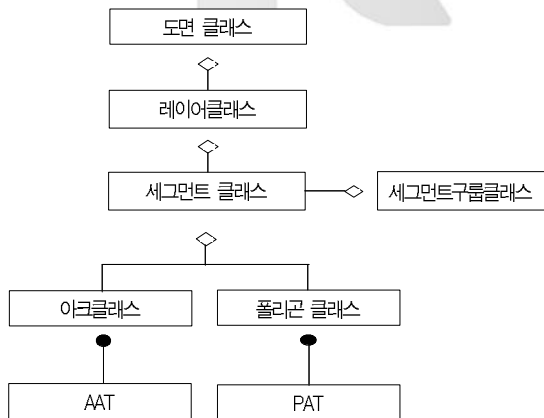


그림 5. class의 계층구조
FIG 5. hierarchy structure of the class

메쉬 단위의 도면 클래스와 레이어 클래스 등을 c++로

정의하면 (그림 6)과 같다.

```
typedef int Layernumber
typedef int meshnumber
class Layer {
    layernumber Layerno;
    string name;
    date malja;
    }
    mesh number meshno;
}
```

그림 6. class 정의
FIG 6. definition of the class

IV. 그래픽 정보의 계층구조

클래스 계층은 중복되는 부분을 최대한으로 줄여서 일반화 관계가 있는 유사한 객체들로 구성할 수 있다. 상위 클래스와 하위 클래스의 관계는 일반화 관계("is a")이므로 하위 클래스는 상위클래스로부터 인스턴스 변수와 메소드 등을 계승받고, 하위 클래스를 정의할 때 상위 클래스 정의에 특성을 삽입하고, 새로 지정하여 다중상속이 가능하다. 도형정보의 상속관계는 (그림 7)과 같다.

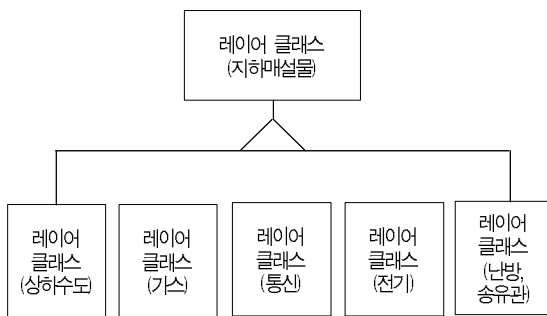


그림 7. 상속계층 구조
FIG 7. hierarchy structure of inheritance

V. 결론

이 논문에서는 J.Rumbaugh의 객체 모델링 기법과 스파이스 참조 모델의 요구사항 변경관리를 토대로 도형정보시스템 구축 틀인 ESRI사의 아크인포를 이용하여 데이터의 추상화와 클래스의 다중상속 표현으로 다양하고 복잡한 도형 정보들의 연관성을 구현하였다. 여러 관련된 객체들을 하나의 논리적 단위로 정의하고 다룰 수 있는 기능을 요구하는 시스템 구축에서 복합객체와 객체의 추상화 표현으로 소프트웨어의 질을 향상시키고 소프트웨어 생명 주기를 효과적으로 지원하며, 풍부한 표기법을 이용하여 시스템의 다양한 측면을 표현할 수 있으므로, 문제 영역을 정확히 모델링하고 유지보수를 용이하게 하므로 소규모 프로젝트보다는 대규모 프로젝트 개발에 사용하면 효과적이라는 결과를 얻을 수 있었다.

전통적인 기존의 개발방법론에서 분석단계에서만 요구사항을 관리하고 다음 단계인 설계, 구현 및 테스트 단계에서 변경관리가 이루어지지 않는 모순점을 해결하기 위하여 스파이스 참조 모델의 요구사항 변경관리와 접목하였으며, 약 20% 정도의 유지보수 비용을 절감할 수 있다고 사료된다.

참고문헌

- [1] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorenson, "object-oriented Modeling and Design," 3th edition, Prentice Hall, 1991.
- [2] 정규장, "웹 기반 공간 데이터 공통 컴포넌트 설계", 한국컴퓨터정보학회 논문지, 제9권, 제1호, 2004
- [3] Fred R. McFadden, Jeffrey A. Hoffer, "Modern

Database Management," McGraw-Hill, 1989.

- [4] 정호원, 황선명, "소프트웨어 프로세스 심사의 이해: SPICE를 중심으로", 정보과학회지, 제17권, 제1호, pp. 6-12, 1999. 01.
- [5] 정창신, 송정범, 이종무, "SPICE 심사사례", 정보과학회지, 제17권, 제1호, pp.13-22, 1999. 01.
- [6] Michael Haug, Eric W. Olsen, and Gonzalo Cuevas, Managing the Change: Software Configuration and Change Management, Springer Verlag, 2001
- [7] 신종철, "요구사항 관리 범위를 위한 명세화 개선 방향", 한국컴퓨터정보학회 논문지, 제6권, 제4호, 2001
- [8] Richard H. Thayer and Merlin Dorfman, Software Requirements Engineering, 2nd Edition, IEEE Computer Society Press, 2000
- [9] Ian Sommerville and Peter Sawyer, Requirements Engineering: A Good Practice Guide, John Wiley & Sons, 1997
- [10] Jan Harrington, "C++ and object-oriented Paradigm," addison-wesley, 1986.

저자 소개

정규장

1980년 동국대학교 전자계산학과

1992년 연세대학교 전산학과

(이학석사)

1996년 충북대학교 컴퓨터학과

(이학박사)

<관심 분야> 소프트웨어공학, 그래

픽알고리즘, internet GIS, 무선통

신 프로토콜, e-러닝, 임베디드 소

프트웨어