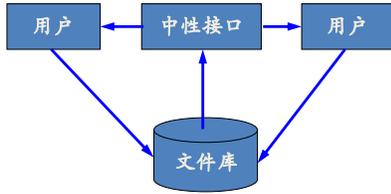


采用中性接口的集成方法

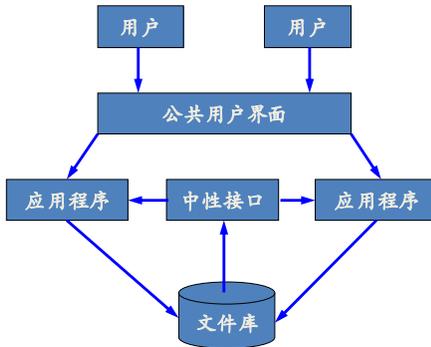


采用公共界面的集成方法

• 采用公共界面的集成方法

- 允许用户在一致的界面下以相同的方式调用不同的应用程序;
- 应用程序存放在文件库中, 数据通过中性接口转换为用户需要的格式;
- 程序和数据有一定的独立性, 但数据还是面向应用;
- 仍存在数据重复、冗余度大、文件不易扩充完善等缺点。

采用公共界面的集成方法



基于工程数据库的集成方法

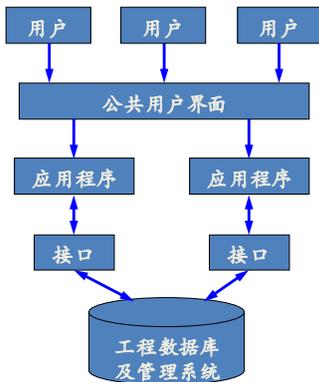
• 基于数据词典的集成方法

- 应用数据词典对多个文件进行存取、控制和管理;
- 具有一定的数据共享和数据的相关性;
- 优缺点介于文件管理相同和数据库集成相同之间。

• 基于工程数据库的集成方法

- 用数据库管理和操作数据简化了信息交换方式;
- 可满足数据共享性、独立性、安全性和完整性;
- 可描述数据本身, 还可通过存取路径描述数据之间的关系;
- 减少了数据冗余度, 节约了空间, 缩短了存取时间。

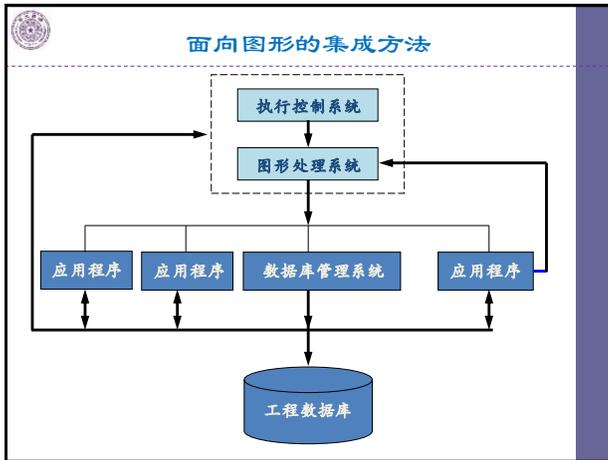
基于工程数据库的集成方法



面向图形的集成方法

• 面向图形的集成方法

- 将集成系统中有关图形处理的部分提取出来独立开发, 启动系统时与执行控制程序一起装入内存, 其他与图形无关的程序均在需要运行时, 由执行控制程序随时调入内存;
- 解决了并行加载问题, 提高了系统运行效果;
- 提高了良好的用户界面, 各子系统可在统一的界面下组织运行系统;
- 图形系统独立与其他部分, 使系统具有良好的可移植性



基于网络的系统集成方法

- **基于网络的系统集成方法**
 - 系统集成是以实现信息的采集、传输、交换、存储、处理与应用的集成化为目标的。
 - **信息集成:** 信息集成是关键, 在于使应用系统加工的对象的各种信息元素实现规范化和体系化, 以便于信息采集、存储、处理和利用。
 - **物理集成:** 物理集成是基础, 解决如何组织应用系统。其任务是技术抉择和产品择优。
 - ✓ **技术抉择:** 确定网络协议集的标准, 即为网络平台确定规范。
 - ✓ **产品择优:** 选择支撑环境, 实现网络平台或通信平台。包括硬件和系统软件。
 - **应用集成:** 应用集成是系统集成的目的, 将相关的应用系统集成一体。

基于中间件的系统集成

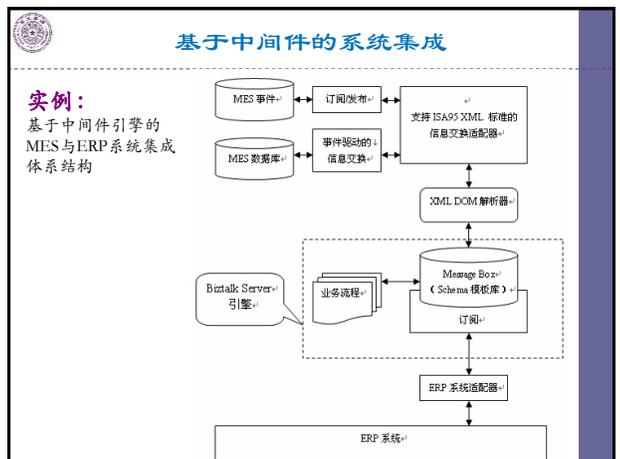
- **基于中间件的系统集成**
 - 为解决分布异构问题, 人们提出了中间件(middleware)的概念, 试图通过屏蔽各种复杂的技术细节使技术问题简单化。
 - 中间件是位于平台(硬件和操作系统)和应用之间的通用服务, 这些服务具有标准的程序接口和协议。针对不同的操作系统和硬件平台, 它们可以有符合接口和协议规范的多重实现。

基于中间件的系统集成

- 中间件在操作系统、网络和数据库之上, 应用软件之下, 总的作用是**为处于自己上层的应用软件提供运行与开发的环境**, 帮助用户灵活、高效地开发和集成复杂的应用软件。
- 中间件是一类软件, 而非一种软件; 中间件**不仅仅实现互连, 还要实现应用之间的互操作**; 中间件是基于分布式处理的软件, 最突出的特点是其网络通信功能。
- **中间件应具有的特点:**
 - 满足大量应用需要。
 - 运行于多种硬件和操作系统平台。
 - 支持分布计算, 提供跨网络、硬件和操作系统的透明性的应用或服务的交互。
 - 支持标准的协议和接口。

基于中间件的系统集成

- 由于标准接口对于可移植性和标准协议, 对于互操作性的重要性, 中间件已成为许多标准化工作的主要部分。
- 对于应用软件开发, 中间件提供的程序接口定义了一个相对稳定的高层应用环境, 不管底层的计算机硬件和系统软件怎样更新换代, 只要将中间件升级更新, 并保持中间件对外的接口定义不变, 应用软件几乎不需任何修改, 从而保护了在应用软件开发和维护中的重大投资。
- 随着计算机软件技术的发展, 中间件技术也已经日渐成熟, 并且出现了不同层次、不同类型的中间件产品。如OMG的CORBA、Sun的J2EE、微软的DNA 2000等。



基于Web Services的系统集成

• 基于Web Services的系统集成

- 随着Internet技术的不断发展, 通过Internet完成软件、服务、客户、供应商、商业流程的互操作、交换以及协作将成为一种发展趋势, 主要体现在以下几个方面:
 - 软件模式的变化: **软件即服务**, 应用软件、套装软件将以一种服务的形式向用户提供;
 - 要求**软件和服务能够通过Internet取得和访问**。
 - 这种面向服务的应用要求一种松散耦合的集成机制, 即它们可以**通过Web随意进行动态交互**。
- Web Services就是顺应这一需求而发展起来的。它是微软提出的基于互联网的开发模型, 是一种软件服务的集成模式。

基于Web Services的系统集成

- Web Services的主要目标就是在现有的不同平台的基础上构筑一个**通用的与平台无关、语言无关的技术层**, 各种不同平台之上的应用依靠这个技术层来实现彼此的连接和集成, 以便为用户提供各种各样的服务。
- Web Service 技术是应用程序通过内联网或者因特网发布和利用软件服务的一种标准机制。
 - 提供了一套分布式的计算技术, 在Internet或Intranet上通过使用标准的XML协议和信息格式提供应用服务。
 - 使用标准的XML协议使得Web服务平台、语言和发布者能够互相独立, 并以一种高度灵活和自动化方式组织交互活动, 建立基础牢固的系统应用集成, 是实现系统集成解决方案的一个理想的选择。

基于Web Services的系统集成

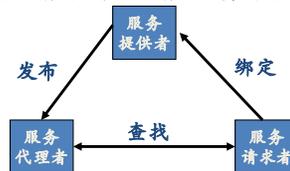
- Web Services特点

- Web Services通过Web为其它应用提供一个定义完善的、模块化的、封装好的功能, 使它们可以很容易地在异构环境中被访问, 从而实现**不同应用程序、系统之间的“松散耦合”集成**。
- Web Services是自成一体、自描述的模块应用程序, 可以通过Web进行发布、查找、调用。一旦配置了Web Services, 其他应用程序(或者是其他Web Services)就可以发现和调用这些服务。
- Web Services独立于特定的编程语言或者操作系统。
- 最重要的是, Web Services依赖于已经存在的传输技术(例如HTTP)和标准数据格式技术(例如XML), 具有与生俱来的标准优势。

基于Web Services的系统集成

- Web Services的基本组成

- **Web Services的基本角色**: 服务请求者、服务提供者、服务代理者。
- **Web Services的角色之间的基本操作**:
 - ✓ **发布**: 服务提供者通过服务代理者注册来配置和发布服务。
 - ✓ **查找**: 服务请求者通过服务代理者查找到服务。
 - ✓ **绑定**: 服务请求者绑定服务提供者, 并使用可用的服务。



基于Web Services的系统集成

- Web Services的实现

- 作为Web Service 服务请求者, 客户程序可以采用UDDI协议发现服务器应用程序(服务提供者)发布的Web Service;
- 采用WSDL 语言确定服务的接口定义;
- 用基于SOAP的XML文档再通过HTTP、FTP和SMTP等常用通信方式交换数据。
 - ✓ SOAP: 简单对象服务协议, 是以XML形式提供的简单、轻量的用于在分散或分布环境中交换结构化和类型信息的机制。能够满足低成本、高效率、易实现、松散耦合、跨平台、与语言无关、与待定接口无关等需求。

基于Web Services的系统集成

- 服务请求者的客户应用程序的在本机调用方法, 会被转换为基于SOAP的XML, 并通过网络发送给服务提供者的应用程序。
- 服务提供者再利用SOAP的XML文档发回对方法调用的响应。
- 由于Web Service 是通过URL、HTTP和XML访问的, 所以运行在任何平台之上, 采用任何语言的应用程序都可以访问XML Web Service。

基于Web Services的系统集成

- Web Services的实施工具

• Microsoft .NET 基础框架平台

- ✓ Microsoft .NET 就是Microsoft XML Web Service 平台。对建立和利用标准的Web Service 提供内置支持。
- ✓ .NET框架内，客户应用程序通过Web Service 侦听器的实现可调用Web Service。
- ✓ 服务器应用程序(服务提供者) 可以用任何 .NET编程语言，比如C#、VB或者托管C++实现Web Service，程序将被编译为微软中介语言MSIL，之后被公共语言运行时CLR 这种虚拟机执行。
- ✓ 客户应用程序(Web Service 用户) 可以用MSXML 或者ASP .NET 实现Web Service 侦听器，以方法调用的方式调用Web Service。

基于Web Services的系统集成

• J2EE 基础框架平台

- ✓ J2EE 由一整套技术和说明规范组成，每一种规范都规定了各类J2EE 函数的操作方式。
- ✓ J2EE 为基于XML的RPC (JAX RPC) 提供了Java API 用以支持面向函数/方法的Web Service 集成。JAX RPC 采用XML 执行远程过程调用(RPC)，同时为汇集参数、传送和接收过程调用提供API。
- ✓ 定义和实现基于JAX RPC 的Web Service。其Web Service 的实现既可以是独立的Java 应用程序，也可以是Enterprise Java Bean (EJB)。JAX RPC API 可以用来创建基于SOAP 的封装器，以确定现有Java 类或者EJB 的WSDL 接口。
- ✓ 在服务器端JAX RPC 运行时，系统上部署Web Service。其部署受到Web Service 具体实现的控制。
- ✓ 客户应用程序通过WSDL 文档说明端口调用Web Service。对客户应用程序来说，对Web Service 的调用应该形如本机方法调用。

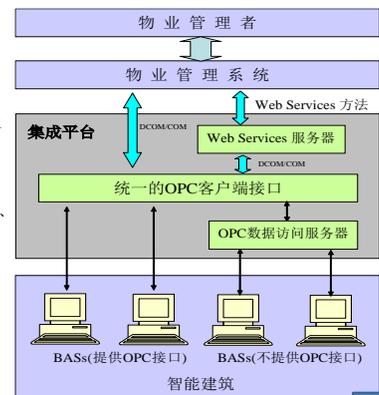
基于Web Services的系统集成

• WebSphere 基础框架平台

- ✓ WebSphere 是IBM 在Web Services 策略中的核心平台。WebSphere 是行业第一个产品化的用于提供和部署Web Service 的功能性基础架构软件平台。
- ✓ WebSphere 支持所有开发、发布、部署Web Service 应用所必须的开放标准和技术，包括UDDI、SOAP、J2EE、WSDL和对XML技术集成的增强。
- ✓ IBM 拥有第一个商业化的开发Web Service 和Java Server Pages (JSP) 的工具组，使程序员能够简单快速地创建、部署和维护基于Web 的应用。

实例： 智能建筑“一体化” 集成模型

- OPC (OLE For Process Control 用于过程控制的OLE) 是一个工业标准。它由一些世界上占领先地位的自动化系统的硬件、软件公司与微软紧密合作而建立的。
- 这个标准定义了应用Microsoft操作系统在基于PC的客户机之间交换自动化实时数据的方法，由国际OPC基金会开发和管理。



9-3 数据交换标准

• 标准输入输出格式

- 采用标准格式输出各系统的设计图形，通过系统的数据转换接口程序，从中提取数据，传递给其他子系统。
- AutoCAD提供的Export命令，可将AutoCAD图形文件转换成各种格式的数据文件，用Import命令将这些数据文件导入到AutoCAD中
 - 3D Studio文件
 - BMP文件
 - DWF文件 (AutoCAD图形文件Web格式)
 - WMF文件 (Windows元文件) ……

标准输入输出格式

- DXF格式 (Drawing Interchange Format)

- AutoCAD的图形交换格式
- CAD图形交换的事实上的标准
- 标准的ASCII文本文件
- AutoCAD中，可利用图形交换命令，交换图形信息。
 - DXFOUT --- 产生DXF交换文件，将AutoCAD图形转换成DXF文件。
 - DXFIN --- 使用DXF交换文件，将DXF文件转换成AutoCAD图形。

产品数据交换标准STEP

- 产品数据交换标准STEP
 - 计算机可理解的关于产品数据表达和交换的国际标准，其目的是提供一种不依赖于具体系统的中性机制，能够描述产品整个生命周期中的产品数据。
 - 使用形式化的数据规范语言EXPRESS来描述产品数据，有利于提供数据表达的精确性和一致性，也有利于STEP的计算机实现。

建筑业国际工业标准IFC

- 建筑业国际工业标准IFC
 - IFC (Industry Foundation Classes) 是由国际协同工作联盟IAI (International Alliance for Interoperability) 为建筑行业发布的建筑产品数据表达标准
 - 提供一个不依赖于任何具体系统的，适合于描述贯穿整个建筑项目生命周期内产品数据的中性机制，可以有效地支持建筑行业各个应用系统之间的数据交换和建筑物全生命周期的数据管理。
 - 统一的数据表达和传输标准。
 - 不同系统之间实现数据交换。

建筑业国际工业标准 IFC

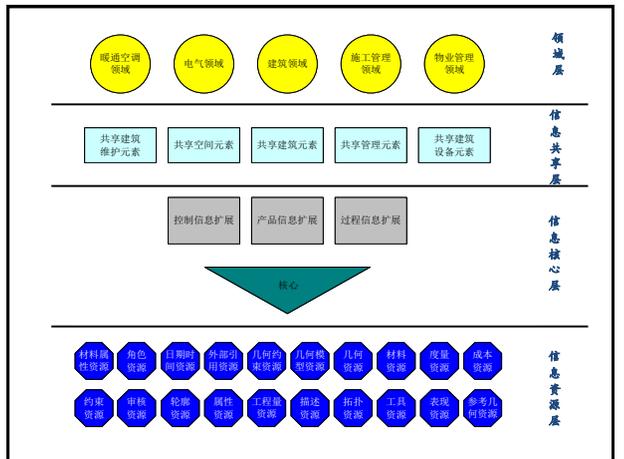
- IFC标准的描述
 - 提供建筑项目工程实施过程所处理的各种信息的描述和定义的规范。
 - 使用“实体” (ENTITY) 作为数据定义的基本元素
 - 通过实体对象的实例组成共享的工程模型，允许不同系统和人员共享模型中的数据。

建筑业国际工业标准 IFC

- IFC大纲 (Schema)
 - IFC大纲是IFC标准的主要内容，其中给出了IFC标准中各种实体的具体数据结构定义。
 - 组成——分为四个层次
 - 信息资源层：定义了标准中用来表示模型的基本信息，如人员信息、属性信息、几何和拓扑信息等。这些基本信息将作为信息描述的基础应用于整个信息模型。
 - 信息核心层：描述建筑工程信息的整体框架，它将资源层的信息用一个整体框架组织起来，使它们相互联系和连接，组成一个整体，真实反映现实世界的结构。核心层中的IfcKernel数据模型提供了代理人 (Proxy)、对象类型 (Object Type) 和属性 (Property) 等定义，因此也是IFC可扩展性的基础。

建筑业国际工业标准 IFC

- 信息共享层：主要用来解决不同领域之间信息交互的问题，在这个层次对各个系统的组成元素细化，定义了一系列的共享元素。
- 领域层：针对建筑行业不同部门领域的特点，深入各个领域的内部，定义了本领域的基本概念，形成各个领域的专题信息。如建筑、暖通空调、电气、施工管理、物业管理等。领域层连同协作层和核心层，提供了用于应用软件信息交换的一系列模型。



建筑业国际工业标准 IFC

- IFC标准的数据定义方式

- 面向对象的规范化数据描述语言EXPRESS。
- EXPRESS语言中语言的定义和对象描述主要用**实体说明**来实现。
 - ✓ 一个实体说明定义了一种对象的数据类型和它的表示符号,它是对现实世界中一种对象的共同性质的描述。类型 (TYPE)、实体 (ENTITY)、规则 (RULE)、函数 (FUNCTION)、过程 (PROCEDURE)
 - ✓ 对象的特性在实体定义中则使用类的属性和规则来表达。实体的属性可以是EXPRESS中的简单数据类型 (数字、字符串、布尔变量等),也可以是其他实体对象。

建筑业国际工业标准 IFC

- ✓ 与其他面向对象语言一样, EXPRESS语言同样可以描述实体之间的继承派生关系。可以通过定义一个实体是另一个实体的子类 (SUBTYPE) 或超类 (SUPERTYPE) 建立实体之间的继承关系, 子类可以继承超类的属性。在IFC标准没有使用多重继承, 所有的实体类型最多只有一个直接超类。
- ✓ 聚合数据类型是基础数据类型的集合, EXPRESS语言可以支持数组 (ARRAY)、包 (BAG)、列表 (LIST)、集合 (SET) 等四种不同的聚合类型。

建筑业国际工业标准 IFC

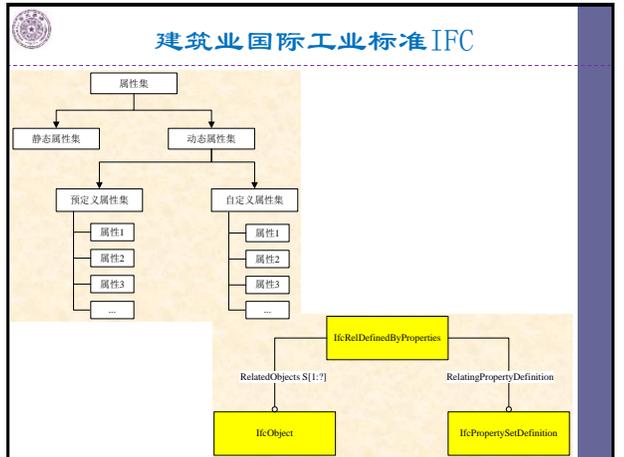
- IFC的信息描述与关联机制

- **基于属性集的信息描述与关联机制**
 - ✓ 属性集的概念

属性集, 顾名思义是属性的集合, 对事物及概念的描述可以通过一条条属性存放于属性集中。属性集提供了一种扩展信息描述的灵活方式。
 - ✓ 信息描述与关联机制

通过IfcRelDefinedByProperties关系实体将IfcPropertySetDefinition描述的信息与IfcObject关联
 - ✓ 适用范围及特点
 - 按需集成
 - 动态扩展
 - 辅助信息描述
 - 易冲突、不宜识别

39



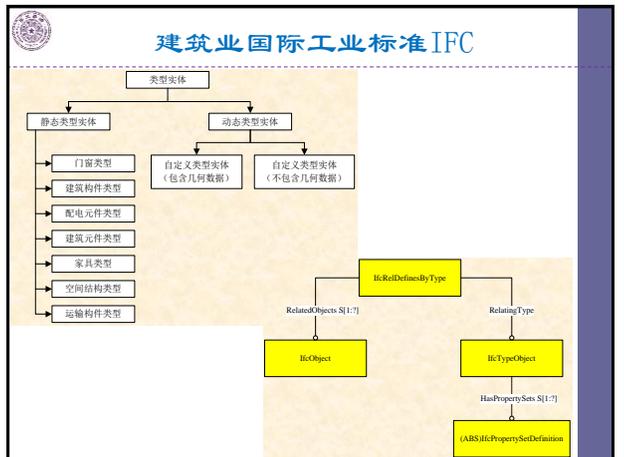
建筑业国际工业标准 IFC

- **基于类型实体的信息描述与关联机制**
 - ✓ 类型实体的概念

类型是具有共同特征的事物所形成的种类。类型实体则是用来存储类型信息的实体。
 - ✓ 信息描述与关联机制

通过IfcRelDefinesByType关系实体将类型信息与IfcObject关联
 - ✓ 适用范围及特点
 - 继承属性集的信息集成特点
 - 动态定义
 - 单一入口
 - 可独立使用

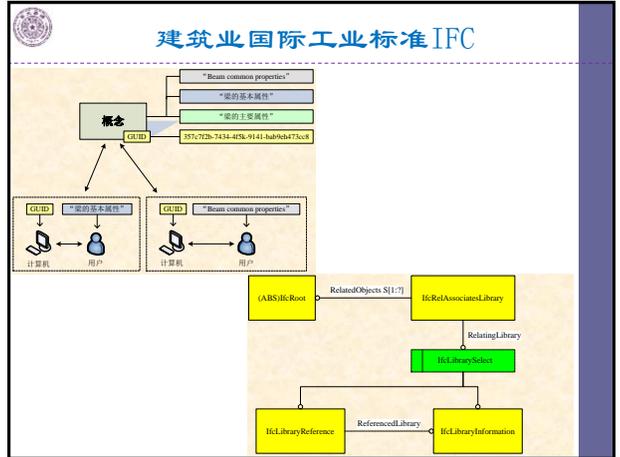
41



建筑业国际工业标准 IFC

- 基于IFD库的信息描述与关联机制
 - ✓ IFD库的概念
 - ✓ 信息描述与关联机制
 - ✓ 适用范围及特点
 - 富语义
 - 概念易识别，不易混淆
 - 扩展性
 - 共享性
 - ✓ 我国IFD库内容的创建途径
 - 通过整理现有的术语定义，获得规范化的术语描述
 - 通过提取建筑规范的相关内容获取属性集定义
 - 通过提取总结建筑企业的工程文档、表格获得常用的属性集，添加到IFD库中

43



建筑业国际工业标准 IFC

IFC的模型信息的扩展机制

- 实体扩展
 - ✓ 扩展机制
 - 基于IfcProxy实体的扩展
 - 通过增加实体定义的扩展
 - ✓ 适用范围及特点

项目	基于IfcProxy实体扩展	通过增加实体定义扩展
易用性	容易	困难
版本兼容性	好	不宜保证
类型安全	类型不安全	采用早联编方式可以提供类型安全
工具箱支持	多数工具箱支持	少数工具箱支持
运行效率	略低	高

- 通过增加大量IFC实体来扩充IFC标准的应用范围。新增实体与原有实体建立关联的方法：
 - 可以通过新扩展实体作为已有实体的子实体；
 - 通过关联实体建立实体。
- 实体扩展按照如下步骤进行：
 - 整理、归纳出所需扩展的实体对象；
 - 定义实体及相关属性信息；
 - 在IFC2x3检索实体对象相关的信息；
 - 建立新增实体对象的派生关系；
 - 建立该实体的关联关系。

建筑业国际工业标准 IFC

属性集扩展

- ✓ 属性集扩展是IFC模型扩展的一种灵活、有效的方式。按照定义方式的不同，属性集分为静态属性集和动态属性集。
 - 静态属性集以IFC实体属性的方式定义，其属性通过IFC大纲静态的定义在属性集中，特定的静态属性集只能应用于特定的IFC实体。
 - 动态属性集则是通过IfcPropertySet实体表示。动态属性集又可分为预定义属性集和自定义属性集。预定义属性集是IFC标准中已经存在的属性集，属性集的名字以“Pset_”为前缀。
- ✓ 自定义属性集为用户根据信息交换的需求自行创建的属性集，在使用过程中需要定义方给出属性集相关信息的约定。
- ✓ 属性信息需要同IFC实体建立关联才能实现信息的描述。属性集通过IfcRelDefinedByProperties关系实体实现IfcPropertySetDefinition与IfcObject建立关联。

IFD (International Framework for Dictionaries)

IFD (International Framework for Dictionaries)

IFD的提出

- IFC中基于属性集的信息描述与关联机制具有易冲突、不易识别的缺点，其根源是属性集采用字符串（Name属性）作为标识。字符串是语义信息的表达，在描述同一概念时具有不确定性，例如同一概念既可由英文又可由中文表达，即使使用同一种语言表达，各地方的习惯用法也不尽相同，另外还存在全称、简称、俗语等多种表达方式。
- 当计算机处理属性集信息时，首先要通过属性集的名称识别属性集。由于属性集名称的不确定性，会出现即使属性集存在，计算机也无法确定的困境，进而影响计算机的自动化处理，造成数据的丢失、冲突。

IFD (International Framework for Dictionaries)

- 1999年，ISO组织为了对信息交换过程中所使用的术语规范开发了相应标准的开发，形成建筑领域面向对象的信息组织框架，标准号ISO 12006-3。
- ISO 12006-3定义了一个语言无关的数据模型，利用该模型可以开发用于存储和提供建筑工程信息的字典，并提供了将信息与分类系统、信息模型、对象模型和过程模型相关联的机制。
- 2006年，正式启动的IFD是基于ISO 12006-3标准建立的术语库。IFD的开发面向全球建筑工程领域，具有开放性、国际化和多语言的特点，作为对IFC的补充和扩展。

IFD (International Framework for Dictionaries)

IFD提供的功能:

- 多种语言的翻译功能;
- 对任何概念的全局唯一标识;
- 对IFC模型扩展, 包括属性集、属性的扩展;
- 存储术语和定义的数据库。
 - ✓ IFD库通过Web Services提供服务, 应用程序可以调用API函数实现对IFD库的访问。
 - ✓ IFC 2x4a版本开始全面支持IFD的使用。目前, 美国、加拿大、挪威、荷兰及日本已经开始针对本国情况建立IFD库。
 - ▶ 我国可采纳IFD现有的技术框架, 并基于该技术框架进行内容开发。然而, 目前我国并没有针对建筑工程领域信息化的专门术语定义及属性集定义, 这些信息尚未被整合归纳, 形成规范标准。

IDM (Information Delivery Manual)

IDM (Information Delivery Manual)

- IDM的提出

- BIM模型的构建贯穿于建筑工程的全生命周期, 是对建筑生命周期工程数据的积累、扩展、集成和应用过程, 是为建筑工程生命周期信息管理而服务的。BIM模型的创建是一个过程, 这个时间跨度长、参与方众多、信息交换复杂。
- 对这个过程的有效管理是实现BIM信息集成的重要基础, 它的实现既需要相应的管理方法, 又需要有效的技术手段。IDM便是针对信息提取与集成过程而提出的一种方法。

IDM (Information Delivery Manual)

- 2006年IAI组织提出了IDM过程控制流程, 用于描述信息交换的过程信息。IDM由3个部分组成:
 - 过程图 (PM, Process Map): 对过程的概述, 包括实现过程的目的, 与之相关的项目阶段以及过程包含的子过程。
 - 交换需求 (ER, Exchange Requirement): 以自然语言描述过程需要的信息及信息源, 以及过程的输出结果。
 - 功能单元 (FP, Functional Part): 从技术角度, 提供过程的技术实现细节, 例如对IFC实体及其属性值的具体要求。
- IDM规范了信息交换过程项目参与方之间的信息流, 提高了信息交换的质量, 支持更加可靠和有效的IFC应用。

建筑业国际工业标准IFC

- “十五”项目: IFC标准研究

- **中国建筑科学研究院:**
 - ✓ 翻译“IFC2x平台规范”;
 - ✓ 建立了PKPM数据文件与IFC的双向数据转换接口。
- **清华大学**
 - ✓ 开发了IFC文件解析器和基于IFC标准的数据接口引擎;
 - ✓ 建立了建筑施工管理的IFC数据描述, 根据IFC大纲 (Schema) 给出的建筑工程信息数据定义, 建立了200余种实体类型。
 - ✓ 实现了建筑结构设计 with 施工信息的交流与共享。

建筑业国际工业标准IFC

- **“十一五”项目: IFC标准研究**
 - **研发了基于IFC的BIM数据集成与管理引擎:** 通过对IFC数据描述规范及解析技术、数据访问及数据转换技术的研究, 完成了自主研发的IFC数据数据集成与管理引擎, 其中包括IFC大纲读取器、IFC文件解析器、IFC数据库访问器等, 实现信息模型的导入和导出和设计 with 施工应用软件的IFC数据的输入输出。
 - **研发了IFC数据库:** 通过研究IFC对象数据模型与关系数据库的映射关系, 完成了可以用于存储海量IFC数据的数据库架构设计和系统开发, 可实现设计与施工BIM信息的存储和管理。
 - **编写建筑施工IFC数据描述标准**

学习重点

- CAE集成化的概念和方法
- IFC标准



提高内容参考

- 开发一个IFC解析器，使其能读取IFC中性文件，并将其内容展现出来，同时具备IFC实体数量统计的功能。



谢谢!

清华大学土木工程系

胡振中

huzhenzhong@tsinghua.edu.cn