

图解说明

序言

“九重宫阙晨霜冷，十里楼台落月明。”
古建筑名类繁多，制式各异，虽万变千幻，亦有迹可循。在此且以《营造法式》与《工程做法则例》为本。

一、宋代材份制与清代斗口制

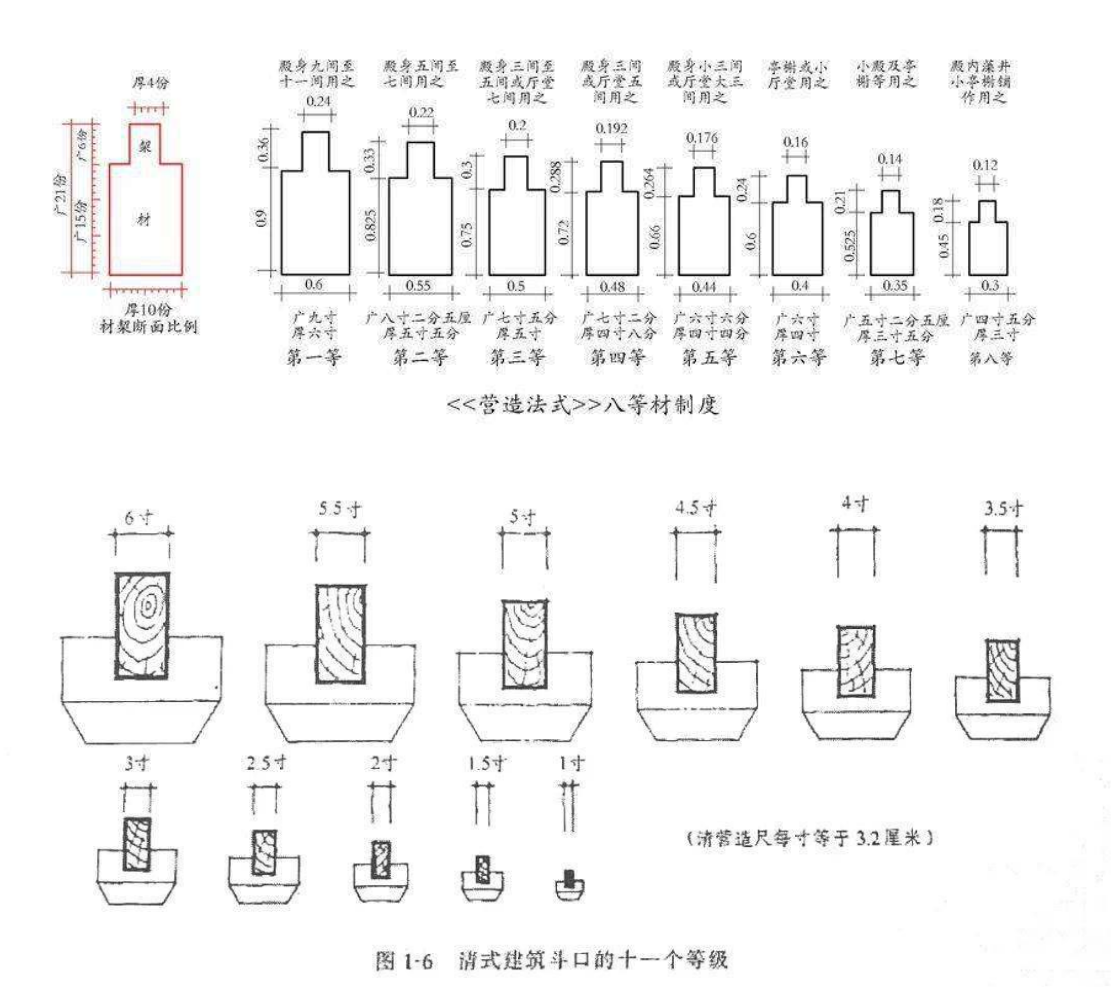
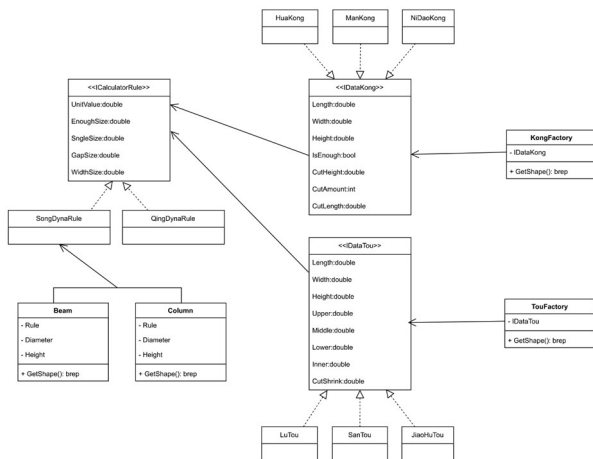


图 1-6 清式建筑斗口的十一个等级

(暂定) 单位换算：宋代 1 尺=10 寸=100 分=30 厘米，清代 1 尺=10 寸=100 分=32 厘米
如图所示，以份或者斗口为度量单位：例如宋代一等材厚 6 寸，十等分，每一份为 0.6 寸，即一个度量单位为 0.6 寸；二等材厚 5.5 寸，同样十等分，每一份为 0.55 寸，即一个度量单位为 0.55 寸，其余同理。虽然等级不同截面尺寸不同，度量单位值随之改变，但是宽高比例是一样的 2：3。清代更加明了，如一等材厚 6 寸，故斗口为 6 寸，即一个度量单位为 6 寸，同理二等材一个度量单位为 5.5 寸。换言之，宋代以断面厚度的十分之一为度量单位，清代以断面厚度为度量单位。

以下简略图示：以度量单位为依据搭建具体模型



```
public interface IUnitConvert
{
    //营造尺单位换算比率，以厘米计：即1尺=? 厘米；
    double Ratio { get; }
    //尺换算成厘米
    double CHI2CM(double chi);
    //寸换算成厘米
    double CUN2CM(double cun);
}

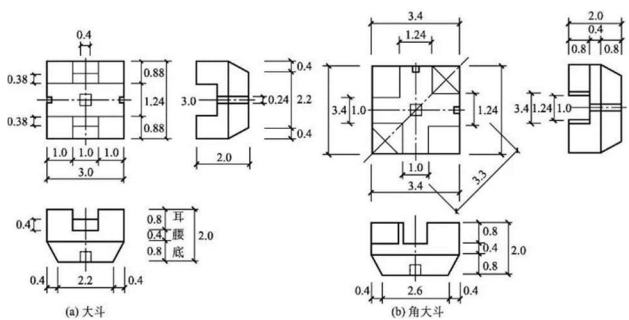
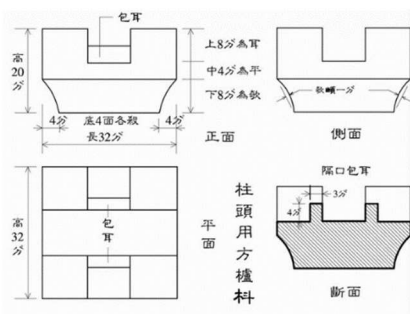
public interface ICalculatorRule
{
    double UnitValue { get; } //度量单位
    double EnoughSize { get; } //足材
    double SingleSize { get; } //单材
    double GapSize { get; } //梁高
    double WidthSize { get; } //材厚
}

public class SongDynaRule : IUnitConvert, ICalculatorRule
{
    //度量单位数值：由用材等级截面尺寸计算得出
    private double _sectionWidth;
    public CaiFenRule(double sectionWidth)
    {
        _sectionWidth = sectionWidth;
    }
    public double Ratio => 30;
    public double CHI2CM(double chi) => Ratio * chi;
    public double CUN2CM(double cun) => CHI2CM(0.1 * cun);
    public double UnitValue => this.CUN2CM(_sectionWidth) * 0.1;
    public double EnoughSize => 21 * UnitValue;
    public double SingleSize => 15 * UnitValue;
    public double GapSize => 6 * UnitValue;
    public double WidthSize => 10 * UnitValue;
}
```

二、斗拱建模分解

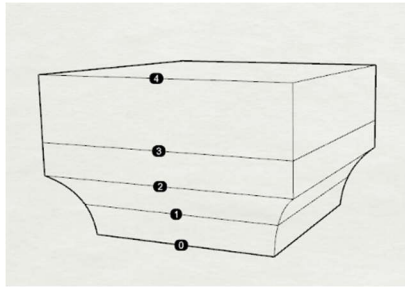
1、数据提取

以大斗为例子，下图左侧为宋式杓料，底部内凹曲线；右侧为清式大斗，底部直线



```
public interface ITou
{
    string TouName { get; } //名称
    double TouLength { get; } //长
    double TouWidth { get; } //宽
    double TouHeight { get; } //高
    double TouUpper { get; } //耳
    double TouMiddle { get; } //平
    double TouLower { get; } //欹
    double InnerOffset { get; } //内凹
    double CutShrink { get; } //底面系分(内缩)
}

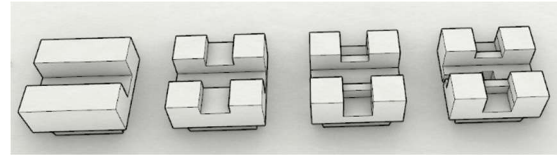
public class LuTou : ITou
{
    private ICalculatorRule _rule;
    public LuTou(ICalculatorRule rule)
    {
        _rule = rule;
    }
    public string TouName { get => "宋式杓料"; }
    public double TouLength { get => 32 * _rule.UnitValue; }
    public double TouWidth { get => 32 * _rule.UnitValue; }
    public double TouHeight { get => 20 * _rule.UnitValue; }
    public double TouUpper { get => 0.4 * TouHeight; }
    public double TouMiddle { get => 0.2 * TouHeight; }
    public double TouLower { get => 0.4 * TouHeight; }
    public double InnerOffset { get => 0.05 * TouHeight; }
    public double CutShrink { get => 0.2 * TouHeight; }
}
```



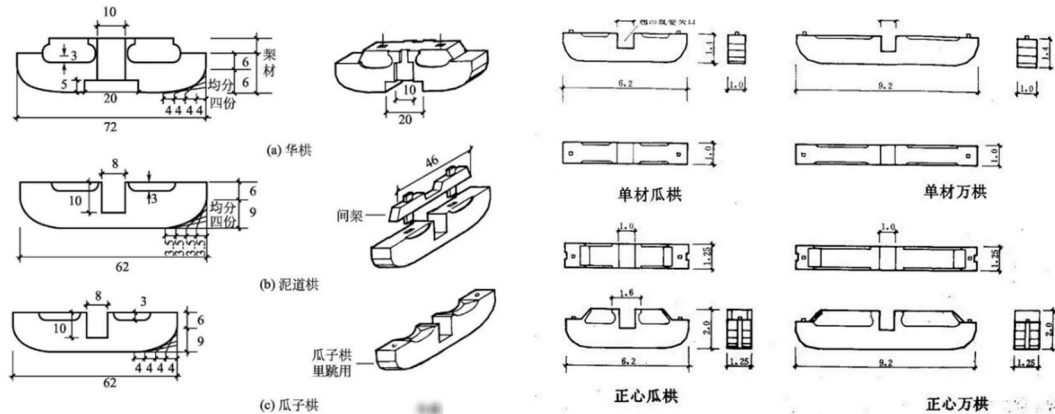
step1: 矩形放样初始实体 S0

step2: 在相应的位置构造实体, 布尔减操作

$S0 \rightarrow S1 \rightarrow S2 \rightarrow S3 \rightarrow S4$;



以华拱为例, 左侧宋式拱, 右侧清式拱

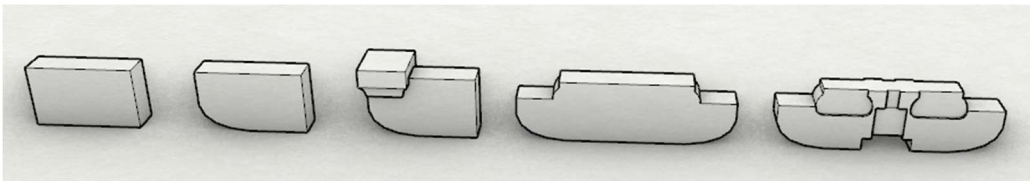


```
public interface IKong
{
    string KongName { get; } //名称
    double ExtendOuter { get; } //外跳长度
    double ExtendInner { get; } //里跳长度
    double KongWidth { get; } //厚度
    double KongHeight { get; } //高度
    bool IsEnough { get; } //是否足材
    double CutHeight { get; } //卷杀高度
    double CutAmount { get; } //卷杀瓣数
    double CutLength { get; } //卷杀总长
    //获取拱的总长: 外跳+里跳+斗底面长度/宽度
    double FullLength { get; }
}

public class HuaKong : IKong
{
    private ICalculatorRule _rule;
    public HuaKong(ICalculatorRule rule)
    {
        _rule = rule;
    }

    public string KongName => "宋式华拱";
    public double ExtendOuter => 31 * _rule.UnitValue;
    public double ExtendInner => 31 * _rule.UnitValue;
    public double KongWidth => _rule.WidthSize;
    public double KongHeight => IsEnough ? _rule.EnoughSize : _rule.SingleSize;
    public bool IsEnough => true;
    public double CutHeight => 9 * _rule.UnitValue;
    public double CutAmount => 4;
    public double CutLength => 16 * _rule.UnitValue;
    double FullLength => ExtendOuter + ExtendInner;
}
```

建模过程同理, 仍然是由初始实体进行一系列操作。



2、对模型制作过程进行封装。看场景需求进行模型继续细化, 或者一个 BOX 足矣。

```
public abstract class TouFactory
{
    protected ITou _dataTou;
    public TouFactory(ITou dataTou)
    {
        _dataTou = dataTou;
    }

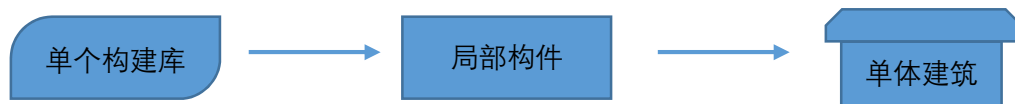
    //初始实体制作过程
    > protected Brep CreateOriginSolid()...
    //可细化初始模型, 或者多分几个私有方法如:
    //protected abstract Brep SomeOtherOperate();
    //.....
    //返回细化后的模型即可
    public abstract Brep GetShape();
}

public abstract class KongFactory
{
    protected IKong _dataKong;
    public KongFactory(IKong dataKong)
    {
        _dataKong = dataKong;
        _position = position;
    }

    > protected Brep CreateOriginSolid()...
    //可细化初始模型, 或者多分几个私有方法如:
    //protected abstract Brep SomeOtherOperate();
    //.....
    //返回细化后的模型即可
    public abstract Brep GetShape();
}
```

3、至于梁柱等其它构件制作基本差不多，然后以某种规则一部分一部分组合起来即可生成整座建筑单体。不那完整的演示视频：

https://www.bilibili.com/video/BV1jf4y1u7G5/?spm_id_from=333.999.0.0&vd_source=8e3794cf0d9dba80bfa2f7a5acb5d48d



三、参考资料

- 陈越 .中国古建筑参数化设计 [D]. 重庆 :重庆大学 ,2002.
- 潘谷西, 何建中 . 《营造法式》解读 [M]. 南京 : 东南大学出版社 ,2005.
- 马炳坚 .中国古建筑木作营造技术 [M]. 北京 :中国建筑工业出版社 ,2003.
- 王茹 .古建筑数字化及三维建模关键技术研究 [D]. 西安 :西北大学 ,2010.
- 唐三元 .明清古建筑参数化三维构件库研究 [D]. 西安 :西安建筑科技大学 ,2012.
- 郭正可 .基于 BIM 的唐代建筑大木作参数化研究 [D]. 太原 :太原理工大学 ,2018
- 清华大学建筑设计研究院 .佛光寺东大殿建筑勘察研究报告 [M].北京 :文物出版社 ,2011