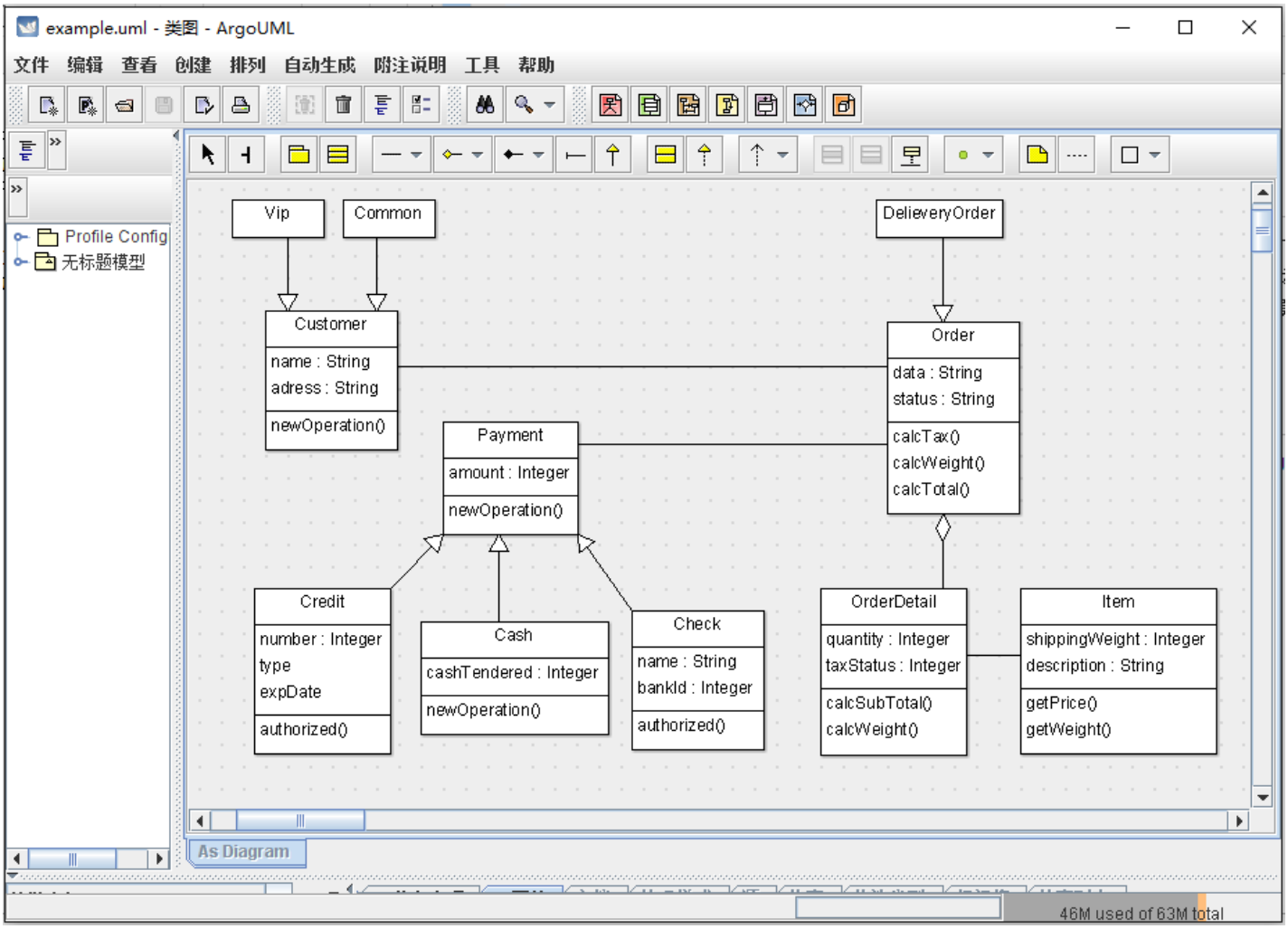
使用说明

1. 概述

本系统是在ArgoUML开源建模的软件基础上开发的，我们对输入的UML类图文件有格式上的要求，它必须是用ArgoUML工具绘制的UML类图并保存成的.uml文件格式(如example.uml)。我们的系统目前具备的功能有（1）查找类图中的环（2）查找类图中任意两个类之间的所有路径（3）就算和获取两个类之间的关系，并给出相应的计算过程。

2. ArgoUML工具介绍

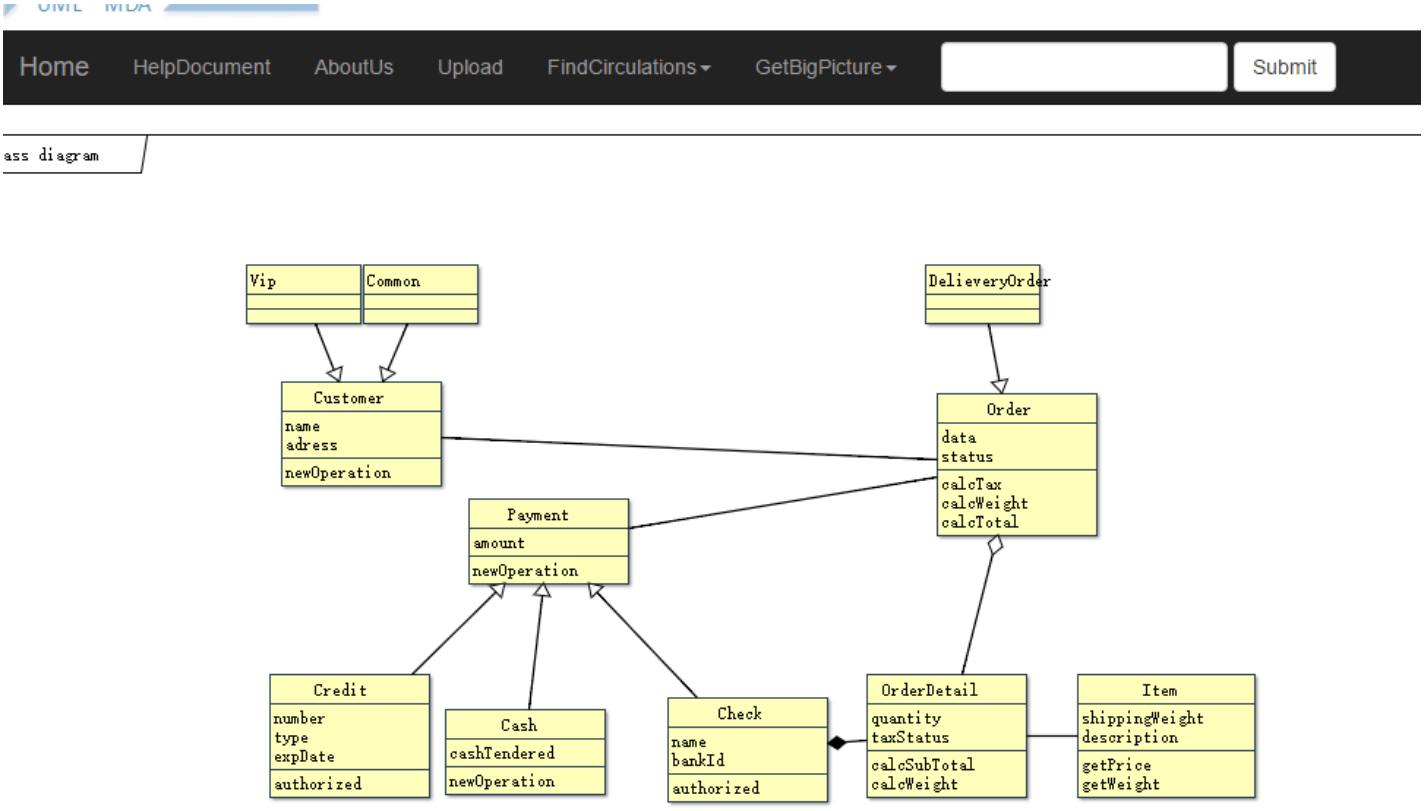
AgoUML与其他的建模工具类似，如Rational Rose，Enterprise Architect， Star UML等，它是一个开源的项目，具备UML建模工具的所有功能。下面是该软件的一个截图。



3. 导入文件

点击Upload进入导入文件页面，从本地选择一个类图（用ArgoUML创建并保存为"xx.uml"格式的文件）

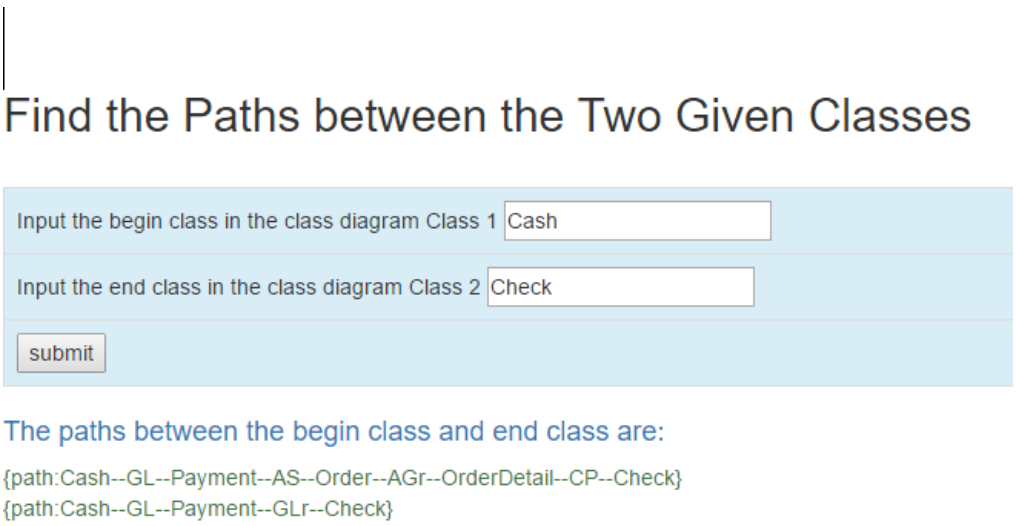
比如我们导入如下的类图，点击导入（文件名为example.uml）



导入后如图所示

4. 获取两个类之间所有的路径

导入完成后，点击FindCirculation菜单下的FindPath，输入类名Cash和类名Check，表示查找这两个类之间的所有路径。结果如下图所示。



5. 查找类图中包含某个类的所有的不重复的环

查找类图中的环十分重要，因为类图中的问题经常出现在环里面。输入类名Check，查找类图中包含Check的环

Find the Circulation Contains the Given Class

Input the class name that to search circulation: Check

submit

The circulations that contains the class are:

{Circulation:Check--GL--Payment--AS--Order--AGr--OrderDetail--CPr--Check}

6. 获取任意两个类之间的关系，并给出计算过程

例如，输入类Payment和类OrderDetail，要得到这两个类之间的关系。对于这两个类之间的这条路径 For path: Payment--AS--Order--AGr--OrderDetail，运用规则AS x Order x AGr equals AS 70，得到Payment--AS--OrderDetail，也就是说这两个类之间是关联关系，可信是是0.7.

Abstract the Paths Between Two Classes and Get the Direct Relationship

Input the begin class Payment

Input the end class OrderDetail

submit

The relationships between the begin class and end class are :

For path: Payment--AS--Order--AGr--OrderDetail
Apply AS x Order x AGr equals AS 70 get: Payment--AS--OrderDetail
Final reliability : 0.7

For path: Payment--GLr--Check--CPr--OrderDetail
Final reliability : 1.0

7. 计算类图中类与类之间关系和相关度矩阵

点击GetBigPicture菜单下的CountCorrelations。进入到计算的页面，在页面中点击Count按钮，得到相关度矩阵，和关系矩阵。（example.uml类图的如下所示）

The correlation matrix is:

The correlation matrix shows the correlation between classes, for example, the element A[i][j] in matrix A means the correlation between class i and class j. Class i is highly correlated to class j if A[i][j]>0.5.

None	Customer	Order	Credit	Item	Payment	DelieveryOrder	Check	OrderDetail	Common	Cash	Vip
Customer	0.0	0.5	0.35	0.175	0.35	0.5	0.35	0.35	1.0	0.35	1.0
Order	0.5	0.0	0.7	0.35	0.7	1.0	0.7	0.7	0.5	0.7	0.5
Credit	0.35	0.7	0.0	0.5	1.0	0.7	1.0	1.0	0.35	1.0	0.35
Item	0.175	0.35	0.5	0.0	0.5	0.35	0.5	0.5	0.175	0.5	0.175
Payment	0.35	0.7	1.0	0.5	0.0	0.7	1.0	1.0	0.35	1.0	0.35
DelieveryOrder	0.5	1.0	0.7	0.35	0.7	0.0	0.7	0.7	0.5	0.7	0.5
Check	0.35	0.7	1.0	0.5	1.0	0.7	0.0	1.0	0.35	1.0	0.35
OrderDetail	0.35	0.7	1.0	0.5	1.0	0.7	1.0	0.0	0.35	1.0	0.35
Common	1.0	0.5	0.35	0.175	0.35	0.5	0.35	0.35	0.0	0.35	1.0
Cash	0.35	0.7	1.0	0.5	1.0	0.7	1.0	1.0	0.35	0.0	0.35
Vip	1.0	0.5	0.35	0.175	0.35	0.5	0.35	0.35	1.0	0.35	0.0

The relationship matrix is:

The relationship matrix shows the direct relationship between classes, for example, the element A[i][j] in matrix A means the relationship between class i and class j. The relationship A[i][j] satisfies the following two conditions.
(1) The reliability of the relationship is the highest
(2) The rank of the relationship is the biggest if the reliability is equal.

None	Customer	Order	Credit	Item	Payment	DelieveryOrder	Check	OrderDetail	Common	Cash	Vip
Customer	None	AS	AS	None	AS	AS	AS	None	GLr	AS	GLr
Order	AS	None	AS	None	AS	GLr	AS	None	AS	AS	AS
Credit	AS	AS	None	AS	GL	AS	None	None	AS	None	AS
Item	AS	AS	AS	None	AS	AS	AS	AS	AS	AS	AS
Payment	AS	AS	GLr	AS	None	AS	GLr	None	AS	GLr	AS
DelieveryOrder	AS	GL	AS	None	AS	None	AS	None	AS	AS	AS
Check	AS	AS	None	AS	GL	AS	None	CPr	AS	None	AS
OrderDetail	None	AG	None	AS	None	AG	CP	None	None	None	None
Common	GL	AS	AS	None	AS	AS	AS	None	None	AS	None
Cash	AS	AS	None	AS	GL	AS	None	None	AS	None	AS
Vip	GL	AS	AS	None	AS	AS	AS	None	None	AS	None

8. 计算类的rank值

类的rank值表示了，在类图中这个类的重要性。rank值越高，表示这个类越重要。点击GetBigPicture菜单下的CountClassRanks。然后进入计算rank的界面，点击CountRank得到rank矩阵。（example.uml的rank矩阵如下所示）

The final class rank matrix is:

The final class rank matrix shows the final class rank of each class. The class is more important if its final class rank is higher. For example, class A is more important than class B if the rank of class A is 200 and the rank of class B is 100

Class	Customer	Order	Credit	Item	Payment	DelieveryOrder	Check	OrderDetail	Common	Cash	Vip
Rank	111.37427	217.01707	74.92005	44.851105	153.851	42.76446	74.92005	55.49429	74.943794	74.92005	74.943794

9. 对rank进行排序

依据rank值大小，从小到大排序。点击GetBigPicture菜单下的SortClass，进入排序页面。然后点击sort按钮得到排序后的rank矩阵。（example.uml的如下所示）

Sort the classes according to rank of classes

SortClass

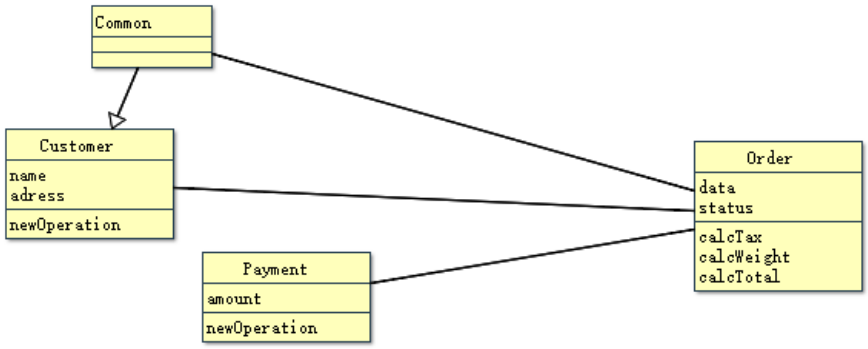
Sort the classes from big to small according to the final class rank:

Class	Order	Payment	Customer	Common	Vip	Credit	Cash	Check	OrderDetail	Item	DelieveryOrder
Rank	217.01707	153.851	111.37427	74.943794	74.943794	74.92005	74.92005	74.92005	55.49429	44.851105	42.76446

10. 获取类图的Big Picture

big picture是类图的抽象，它包含若干类图中的重要的类。通过重要的类来了解类图的设计。 点击GetBigPicture菜单下的GetBigPicture，进入获取big picture的页面。

1. 可以设定抽象的层次（5层）
   如输入，4，点击提交按钮。（对于example.uml如下所示）



2. 可以设定保留几个类

如输入2，保留两个重要的类（对于example.uml如下所示）

3. 系统自动抽象
点击默认的他提交按钮（对于example.uml如下所示）