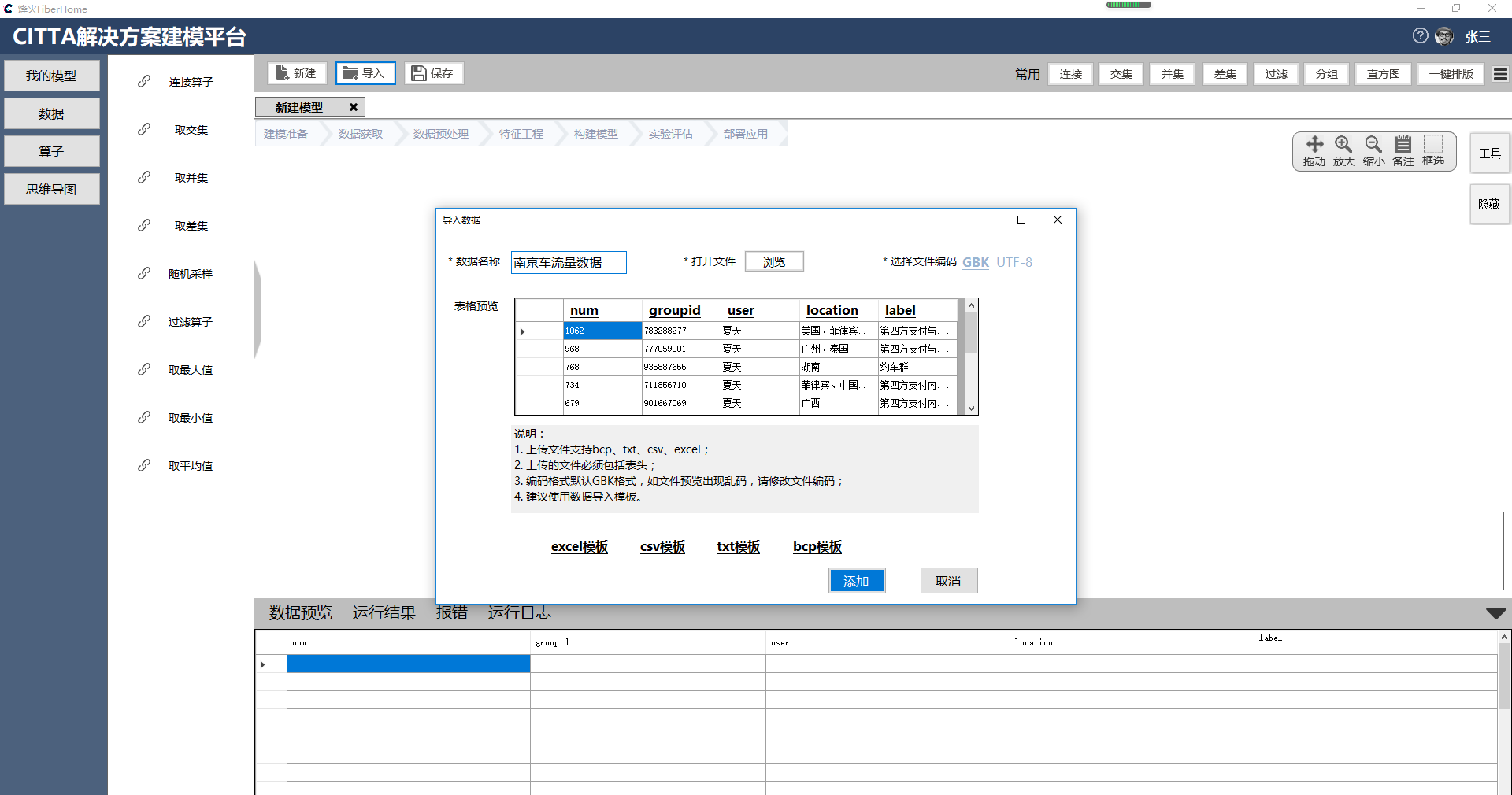
1用户模型文档的数据保存

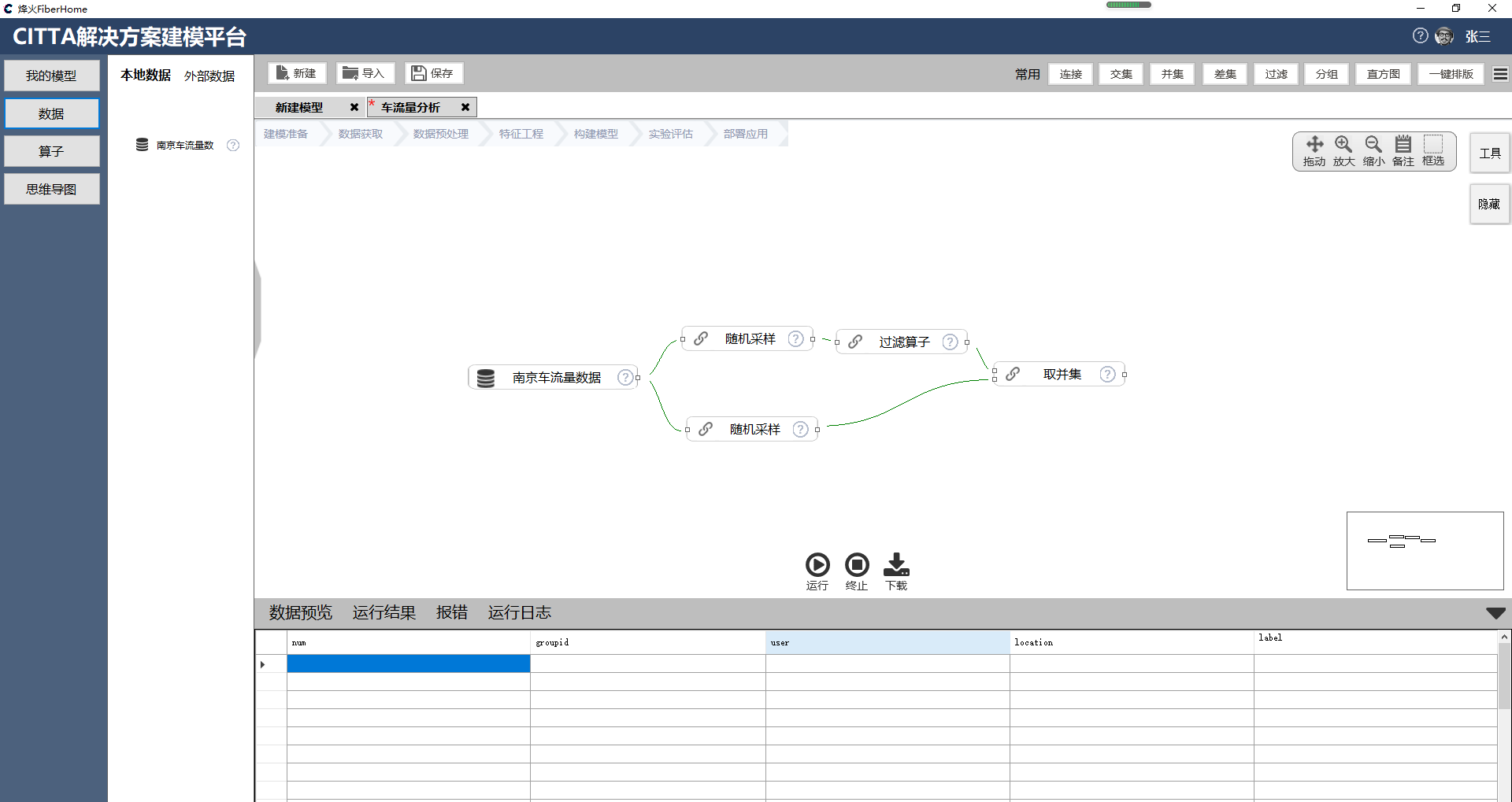
**1.1功能描述**

保存用户创建的模型文档中所有元素，用户再次登录，该用户的模型文档所有数据自动加载。

**1.2界面**



导入数据源

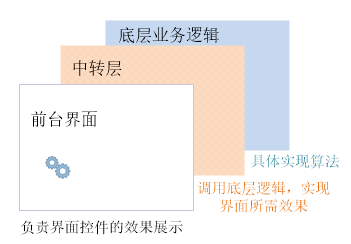


新建模型文档

**1.3业务操作说明**

1. 点击”导入”按钮，导入待分析的数据源，在左侧“数据”选项的“本地数据”，出现导入的数据源模块。
2. 将数据源模块拖拽进入模型文档，在“算子”选项中拖拽用于数据分析的各个算子进入模型文档，模型文档内容被更改，模型文档标题会出现“\*”。
3. 右侧“浮动工具栏”中“备注”选项可添加用户构建的模型备注信息，备注信息更改，模型文档标题会出现“\*”。
4. 点击不同的模型文档标题，进行文档切换，界面显示当前文档中的元素。
5. 点击模型文档标题的“×”可以关闭当前文档，但是界面删除到剩下最后一个文档就不再删除，以保留该显示栏中始终包含一个模型文档标题。
6. 点击“保存”按钮，保存该用户模型文档中所有数据，“我的模型”工具栏会出现对应的模型文档的图标。
7. 用户重新登录，已保存的所有模型文档内容自动加载，默认当前打开的模型文档为最后加载模型文档，“我的模型”栏加载对应的模型文档图标。

**1.4业务规则说明**

****

业务实现的代码分层设计

**1.4.1用户登录界面**

一、详细功能设计：

登陆界面加载时，读取保存历史用户信息与最后登录用户名的用户信息文件，若文件不为空，登陆界面的用户名输入框显示最后登录的用户名，其下拉列表显示历史登录用户名。界面点击登陆按钮，如果用户信息文件不包含登陆者的用户名，则向用户信息文件写入该用户相应信息。

二、登录用户信息持久化设计主要思想：

* 1. 登录的用户信息存储到某个路径下的xml文件。例如：D:\work\Citta \Citta\_T1 \bin\Debug \cittaModelDocument\UserInformation.xml。
  2. 登录界面加载时，从xml文件加载用户信息
  3. 用户分历史用户和最后登录用户，因而文档内容设计为一下格式：

<login>

<user>张三</user>

<user>张思</user>

<lastlogin>张思</lastlogin>

</login>

* 1. 当有新用户名在界面登录时，xml文件增加一个<user>字段记录用户名，<lastlogin>字段内容更新为当前用户名。

**1.4.2 主界面**

一、详细功能设计：

1. 界面加载：读取该用户保存的历史模型文档。无模型文档，界面自动新建一个模型文档；存在模型文档，左侧“”我的模型“”中加载相应模型文档图标，与此同时界面加载所有模型文档及文档中的内容，默认界面打开最后一个加载的模型文档（模型文档标题框为黑色，标识当前打开文档），打开的模型文档中元素show,其他模型文档元素hide。其中，若用户删除或变动了数据源的位置，则左侧“数据”栏的“本地数据”中对应数据源模块出现黄色叹号，模型文档中数据源元件出现黄色叹号
2. 数据源导入：点击”导入”按钮，在左侧“数据”选项的“本地数据”，显示导入的数据源，同时在D:\work\用户名\LocalData\datasource.xml（文件不存在则新建该文件）内按行存储数据源信息，存储格式——数据源名称：数据源文件存储路径。若在数据源上点击右键，选择删除，则该数据源模块消失，D:\work\用户名\LocalData\datasource.xml中对应数据源记录清除。
3. 新建模型文档：点击“”新建”按钮，出现新建模型文档的弹窗提示，点击“弹窗中的添加”按钮，读取该用户目录下保存的模型文档名称，如果存在命名重复情况，出现请用户重命名的弹窗警告，不存在命名重复情况，则直接新建；点击“取消”按钮，返回。
4. 模型文档内容发生变化：从“算子”栏目”和“数据”栏，拖动算子到界面，当前文档用”\*”标记已更改，对更改的内容进行步骤（6）的保存，”\*”标记消失。
5. 模型文档关闭：点击模型文档标题栏“叉号”，若文档已保存，该文档关闭；若文档未保存出现弹窗警告，以确认是否保存，点击“不保存”直接关闭文档；点击“取消”仅弹窗关闭；点击“保存”按钮，操作如步骤(6)。如果界面删除到剩下最后一个文档就不再删除，以保留该显示栏中始终包含一个模型文档标题。
6. 模型文档保存：点击“保存”按钮，若模型文档未修改直接关闭，否则判断 “D:\work\用户名\模型文档名称\模型文档名称“.xml“路径是否存在。若不存在，直接创建该路径，并将模型文档内容写入模型文档名称“.xml“；若存在，则覆盖重写“D:\work\张三\模型文档名称\模型文档名称.xml”文件内容。模型文档标题移除。
7. 快捷菜单栏中“”保存“”：点击该“”保存“按钮”，若当前文档未修改，直接返回，否则判断 “D:\work\用户名\模型文档名称\模型文档名称“.xml“路径是否存在。若不存在，直接创建该路径，并将模型文档内容写入模型文档名称“.xml“；若存在，则覆盖重写“D:\work\张三\模型文档名称\模型文档名称.xml”文件内容。
8. 界面关闭：点击主窗体右上角“叉号”，内容为空未保存的文档直接关闭，内容不为空没有保存的文档，弹窗警告有未保存的文档，点击“取消”按钮，界面取消关闭，仍停留在当前界面，点击“关闭”按钮，直接关闭文档不保存，点击“保存”按钮，转到步骤（6），对文档保存。
9. 模型文档切换：点击其他文档标题，当前文档中所有算子、数据源、备注等对象属性设置为“隐藏(Hide)”，激活的新文档中包含的所有算子、数据源、备注等对象属性设置为“出现(Show)”，以此实现文档内容切换。

二、用户空间信息持久化设计主要思想：

用户空间展示的主体内容分为两类，一是数据源，二是模型文档。因而用户空间信息持久化主要是保证该用户空间下数据源信息持久化，与模型文档信息持久化。

数据源持久化方面：

1. 当用户空间导入本地（数据库）数据源时，就需要将数据源的信息保存在该用户名路径下的xmL文件，文件中保存的是重现数据源的关键信息。
2. 数据源的存储路径设计为“初始路径\用户名\本地数据源（外地数据源）\数据源名称.xml”。例如：D:\work\Citta\Citta\_T1\bin\Debug\cittaModelDocument\张三\LocalData\datasource.xml。
3. 保存数据源关键信息的xml文件内容格式设计为：

<DataSourceDocument>

<DataSource>

<name>XX数据源</name>

<code>UTF-8</type>

<path> D:\c#coderaw\趋势数据.txt </subtype>

< frequency >3</ frequency >

</DataSource>

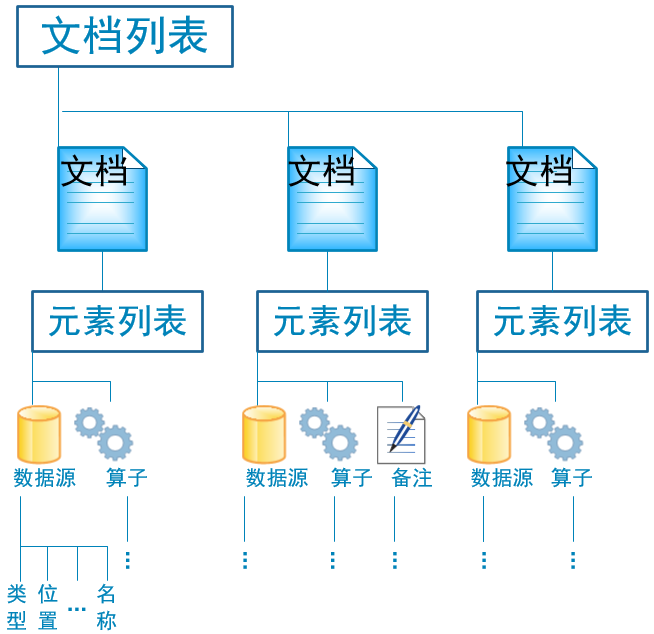
</DataSourceDocument>

<name>字段存储导入的数据源名称，<code>字段存储数据源的编码格式，<path>字段存储数据源的路径，以保证找到路径加载数据，如果该路径中文件为空，即被用户删除，则加载后的数据源图标会被警示图标标记（如黄色叹号）， < frequency >存储数据源在该用户的各个模型文档中的引用次数，引用次数为0，用户可对该数据源进行删除操作，不为0用户则不能删除该数据源。

1. 用户导入新的数据源，xml文件就会增加一个<DataSource>节点，并在其下对应的字段中写入复现数据源的关键信息，用户删除数据源，对应的<DataSource>节点所有内容删除，用户操作引起的数据源的一些关键信息的修改，<DataSource>节点内对应的内容也会被更新。

模型文档持久化方面：

1. 用户新建模型文档，先将反应文档的关键信息写入内存；用户在模型文档中增加算子、数据源、文档备注等元素时，反应这些元素的关键信息也会同时写入内存，并且要关联其关联该模型文档。
2. 因而将一个模型文档定义为一个Document类，模型文档中一个元素定义为ModelElement类，由此模型文档存储层级设计为：



模型文档存储层级图

1. 用户界面中各模型文档内容的修改，如模型文档中元素的增删改操作，以及模型文档之间的切换，模型文档列表中相应模型文档对象也会对相关数据进行修改，模型文档元素的改变以及文档切换等效果的实现都会在底层业务中书写，从而建立界面效果与底层逻辑实现的对应关系。
2. 底层业务逻辑实现中，需要明确的界面中引起交互动作的事件点有：

算子、数据源拖入canvasPanel, 触发文档模型新增元素事件，当前文档dirty；

算子、数据源重命名，触发文档模型元素修改事件，当前文档dirty；

算子、数据源位置改变，触发文档模型元素修改事件，当前文档dirty；

算子、数据源被删除，触发文档模型元素修改事件，当前文档dirty；

备注写入内容，触发备注内容修改事件，当前文档dirty；

备注内容删除，触发备注内容修改事件，当前文档dirty；

数据源拖入canvasPanel，触发数据源引用次数修改事件；

数据源删除canvasPanel，触发数据源引用次数修改事件；

模型文档保存，触发模型文档内容写入事件；

模型文档保存后删除，触发界面/缩略框中该模型文档元素删除事件；

模型文档删除，触发界面、缩略框中该模型文档元素删除事件；

Click模型文档标题，触发文档模型切换事件；

1. 用户保存当前模型文件时，会在该用户文件中保存对应模型文档xml文件。
2. 模型文档存储路径设计：“初始路径\用户名\模型文档名称\模型文档名称.xml”。例如：D:\work\Citta \Citta\_T1\bin\Debug\cittaModelDocument\张三\车辆模型\车辆模型.xml。
3. 每个存储模型文档信息的xml文件内容格式设计为：

<ModelDocument>

<ModelElement>

<name>取并集</name>

<type>Operate</type>

<subtype>UnionOperator</subtype>

<location>{X=355,Y=133}</location>

<status>Null</status>

</ModelElement>

<ModelElement>

<name>取平均值</name>

<type>Operate</type>

<subtype>MeanValueOperator</subtype>

<location>{X=450,Y=213}</location>

<status>Null</status>

</ModelElement>

</ModelDocument>

<name>为元素类型，<type>元素类型，主要有算子类型，数据源类型，备注类型等， <subtype>为子类型，如算子类型中的取并集类型、取平均值类型、随机采样，<location>为算子的位置，<status>为元素的状态，如是否运行过，是否生成结果等。

1. 用户再次登录，就会遍历该用户名文件下的所有模型文件，根据xml文件中记录的元素的各个属性，新建该元素并重载到界面。