Lerisk－i* 框架自动建模与编辑工具介绍

李天颖 刘 璘 寇晓溪 赵德旺
（清华大学软件学院 北京 100084）

摘 要 在需求工程中，基于主体的i* 建模框架（主要包括策略依赖模型及策略推理模型）已经成为最常用的早期需求建模与分析的工具之一，而且关于i* 建模框架的编辑工具开发也有很多相关的研究工作。然而现有的这些工具往往只提供诸如模型图编辑、存储等基本功能，而笔者需要在需求工程小组的项目中为对需求文本进行建模的结果进行模型可视化，同时提供编辑存储及自动布局功能，并开发出新的基于i* 建模框架的工具。文中首先对主流的i* 建模工具进行了调研，研究了建模工具的基本功能，同时分析了其功能的不足点，在此基础上提出了新工具设计的功能补充点；然后对i* 框架的布局问题进行介绍并详细描述了其自动布局算法的实现，给出了可视化工具的详细设计；最后在此工具的基础上，进行了实际需求文本的建模及模型编辑功能的实验，并将此工具与主流工具的功能进行对比，以展示本工具的功能特点。

关键词 i* 建模框架，可视化工具，自动布局，策略依赖，策略推理
中图法分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.09.007

Introduction of i Star Framework Auto Modeling and Edit Software Tool "Lerisk"

LI Tian-ying LIU Lin KOU Xiao-xi ZHAO De-wang
(School of Software, Tsinghua University, Beijing 100084, China)

Abstract In the field of requirements engineering, the subject based on *i** framework including the strategy dependency model and strategy rationale model has become one of the most commonly used tools in the early requirements modeling and analysis. At the same time, many research works focus on the development of *i** modeling framework editing tools. However, the existing tools often provide only basic functions, such as model diagram editing, storage, etc., which do not satisfy the requirements that we need to visualize the modeling results of requirements text in our requirements engineering group, and also provide the editing storage and the automatic layout. Therefore, it is needed to develop a new tool based on *i** framework. Firstly, this thesis investigated and surveyed the well-known modeling tools and analyzed the basic functions of them, as well as the shortcomings. Then, on this basis, we proposed and designed the supplement functions of our new tool, and introduced the layout problems of *i** framework, as well as detailedly described the implementation of its automatic layout algorithm. Finally, we presented the detailed design of the visualization tool. Based on the proposed tool, we carried out experiments on the requirements text modeling and the model editing functions, and also compared them with the corresponding functions of well-known modeling tools to demonstrate the features of our tool.

Keywords *i** framework, Visualization tool, Automatic layout, Strategy dependency, Strategy rationale

1 引言

1.1 研究背景

在需求工程[1]中，基于主体的i* 建模框架（主要包括策略依赖模型及策略推理模型）已经成为最常用的早期需求建模与分析[2-4]的工具之一，而且关于i* 建模框架的编辑工具开发也有很多相关研究工作。然而现有工具往往只提供诸如模型图编辑、存储等基本功能，并且以桌面程序的形式被使用，跨平台性、扩展性、通用性和用户体验都不够好。笔者在需求工程小组的项目中需要对需求文本进行建模的结果进行模型可视化，同时提供编辑存储及自动布局功能，因而需要开发出新的基于i* 建模框架的工具。工具首先对输入的自然需求文本进行处理，分析出其中的策略依赖模型关系，然后进行图形可视化。然而其并没有对于策略依赖模型进行自动布局的成型算法[5]，本文实现了该算法并将其应用到该工具中。

在工具进行自然语言处理的基础上，用户可以对模型进

到稿日期：2013-12-25 返修日期：2014-01-16 本文受国家基础研究发展计划（973计划）（2009CB320706），国家高技术研究发展计划（863计划）（2012AA040904），国家自然科学基金重大项目（90818026）资助。

李天颖（1988－），女，硕士生，主要研究领域为需求抽取与建模，E-mail：litianying10@mails.tsinghua.edu.cn；**刘 璘**（1973－），女，博士，副教授，主要研究领域为软件需求工程、信息系统工程、安全工程及知识管理；**寇晓溪**（1982－），男，硕士生，主要研究领域为需求抽取与建模工具开发；**赵德旺**（1990－），男，主要研究领域为自然语言处理。

行再编辑、保存、导入导出等操作，或者直接新建模型图再进行编辑。因此工具需要实现 i* 建模框架的基本编辑功能，同时提供对创建或修改的模型图进行再自动布局的功能。为了方便地与小组内其它人员及其它工具系统进行信息交流，工具对于模型图的数据结构及模型图的存储文件格式进行了标准化设计，为外界的模型再处理提供了合适的接口。

1.2 相关知识

i* 建模框架[1]基于策略主体，被广泛用来进行早期的需求建模和分析。它的命名意味着“分布式的多重意图”[9,10]。i* 为支持早期的需求工程分析、理解工程的环境、调研不同的系统提供了一个需求建模的框架，同时它可以用来分析系统对组织中主体的影响[6-8]。i* 建模框架由两部分组成：策略依赖模型和策略推理模型。

策略依赖模型(Strategic Dependency，SD)：主要是强调主体之间交互活动背后的意图和动机，它描述了主体间社交依赖关系的网络。策略依赖模型由节点和连接组成。节点表示一个主体，两主体之间的连接表示一个主体依赖对方实现某个目的。前者称为依赖者(depender)，后者称为被依赖者(dependee)，其依赖的事物为依赖物(dependum)。其依赖类型分为 4 种，分别为：目标依赖(goal dependency)、任务依赖(task dependency)、资源依赖(resource dependency)及软目标依赖(softgoal dependency)。图 1 为一个策略依赖模型图示例。

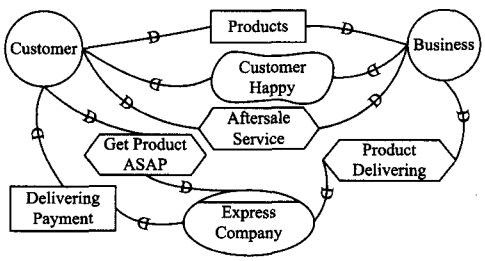


图 1 策略依赖模型图

策略推理模型(Strategic Rationale，SR)：描述了主体对现有的过程和其他候选过程的分析和选择评价，以及它们是如何进行目标分解、目的手段推理和针对软目标的贡献网络来实现的。图 2 为策略推理模型图。

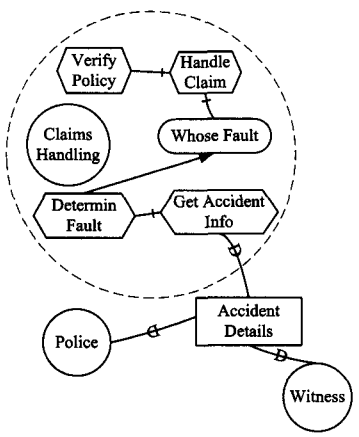


图 2 策略推理模型图

1.3 文章结构

本文第 1 节先对主流的 i* 建模工具进行了调研，研究了建模工具的基本功能，同时分析了其功能的不足，在此基础上整理了建模工具的基本功能并提出了新工具设计的功能补充点，然后对 i* 框架的相关知识进行了介绍；第 2 节详细介绍了工具的设计与实现；第 3 节介绍了工具的实验结果并与主流工具进行功能对比；第 4 节对本文内容进行了总结，并对以后的工作进行展望。

2 可视化工具的设计与实现

2.1 工具功能及架构

2.1.1 工具概述

本文主要工作是实现一个可视化工具，它可以对自然需求文本进行模型处理，并以可视化的形式展示出来，同时提供编辑保存等功能。根据第 1 节中对主流工具的调研，并考虑到扩展性和跨平台性，这里提出该可视化工具的主要功能，图 3 为该工具的主要功能用例图。本工具命名为“Lerisk”。

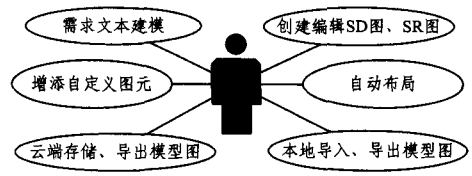


图 3 工具 Lerisk 主要功能用例图

2.1.2 系统架构

考虑到 Lerisk 必须具有良好跨平台性和用户体验，同时支持云存储及轻便性，我们使用 Flex 技术进行客户端的开发，使用 Java 进行服务器端的开发，使用 Blaze Data Service 技术进行通信。Lerisk 的系统模块架构如图 4 所示。

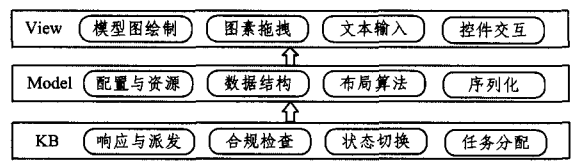


图 4 Lerisk 的系统架构

开发环境：

系统：Window 7 32 位

服务器端开发 IDE：Eclipse (Galileo)

客户端开发 IDE：Adobe FlashBuilder4. 6 Premium

运行环境：Google Chrome 浏览器

2.1.3 系统主要功能介绍

1. 创建编辑策略依赖模型图

介绍：用户可以创建空白的策略依赖模型图或在导入的模型图基础上进行编辑，以下为 i* 建模框架编辑工具的基本功能。

子功能：

- 创建空白策略依赖模型图
- 添加策略依赖模型可选的 4 种主体元素(ActorElement)
- 添加策略依赖模型可选的 4 种意图元素(Intentional Element)
- 添加可选的 10 种依赖连接(Link)
- 连接合规性检查
- 修改元素显示的文字信息
- 调整元素的宽度和高度
- 拖拽摆放元素

· 拖拽和缩放模型图
· 删除元素和连接
· 切换直线/曲线形式连接
· 修改连接线颜色
· 选中元素(光晕)
· 选中连接(光晕)
· 修改光晕颜色。

2. 创建编辑策略推理模型图

介绍:用户可以创建空白的策略推理模型图或在导入的模型图基础上进行编辑,以下亦为 i* 建模框架编辑工具的基本功能。

子功能:

· 创建空白策略推理图
· 添加策略推理模型可选的 1 种主体元素(SRActorElement)
· 添加策略推理模型可选的 4 种意图元素(IntentionalElement)
· 为策略推理主体元素添加意图子元素
· 解除意图子元素的从属关系
· 添加可选的 10 种依赖连接(Link)
· 连接合规性检查
· 修改元素显示的文字信息
· 调整元素的宽度和高度
· 拖拽摆放策略推理主体元素并同时移动其从属子元素
· 拖拽摆放意图元素
· 拖拽和缩放模型图
· 删除策略推理主体元素的同时删除其从属子元素
· 删除意图元素和连接
· 切换直线/曲线形式连接
· 修改连接线颜色
· 选中元素(光晕)
· 选中连接(光晕)
· 修改光晕颜色。

3. 模型图的本地存储及导入

介绍:用户可以对正在编辑的模型图进行本地存储,存储为 xml 文件,并可以在任意客户端上导入此文件以继续编辑或展示。

子功能:

· 选择本地保存路径
· 自定义文件名
· 本地保存成 xml 文件
· 选择要打开的本地文件
· 导入该文件并绘制出该模型图。

4. 模型图的云存储及云导入

介绍:用户可以对正在编辑的模型图进行云存储,并可以在任意客户端上远程导入此文件以继续编辑或展示。

子功能:

· 定义云存储文件名称
· 检查名称是否已经存在并提示
· 按照该名称进行云存储
· 输入要远程导入的文件名
· 云端导入该文件。

5. 增添自定义元素

介绍:用户可以通过自定义的形式来扩充元件,为元件指定名称和图标,其功能性质和原始各元素完全一致。

子功能:

· 定义新元素的名称
· 定义新元素的类别、主体元素/意图元素
· 上传元素图标
· 增添该元素
· 在模型图编辑中加入该元素,其具有相应类别元素的所有性质。

6. 需求文本建模

介绍:用户通过提交需求文本来对其建模,并将建模后信息展示出来。

子功能:

· 打开要处理的文本文件
· 输入要处理的文本文字
· 提交并进行模型处理
· 将处理后的模型绘制在面板上。

7. 自动布局

介绍:对于用户已经编辑的策略依赖模型图,可以使用自动布局功能来进行排版从而得到一个比较美观的模型图。

子功能:

· 对已有的策略依赖模型图进行自动布局
· 将布局结果显示在面板上并提示是否有孤立的元素。

2.2 系统实现

2.2.1 界面介绍

图 5 为 Lerisk 主界面的展示,主界面主要分为单工具栏、控件栏、属性栏及绘图面板 4 个部分。

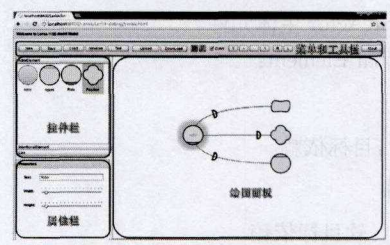


图 5 Lerisk 的界面展示

1. 菜单工具栏

菜单工具栏主要为文件操作及模型编辑提供快捷的操作,由以下模块构成:

New:新建一个策略依赖或策略推理模型图。

Save:本地保存已经编辑的模型图

Load:加载本地模型图文件

Newicon:增添自定义元素

Text:为自然需求文本进行建模处理

Upload:云端保存已经编辑的模型图

DownLoad:云端加载已经保存的模型图

 :调整连接线颜色

 :调整选中元件光晕颜色

☑ Curve：切换连接线的直线/曲线模式

×：删除选中的元件

+：放大模型图(同鼠标滚轮向上功能)

—：缩小模型图(同鼠标滚轮向下功能)

1：原始大小

R：随机布局

L：自动布局

About：关于 Lerisk 信息

2. 控件栏

控件栏主要用来放置需要添加到模型图中的元件，包括主体元件(Actor Element)、意图元件(Intentional Element)及连接(Link)3 种。意图元件默认有 4 种，连接有 10 种，而主体元件在策略依赖模型图中有 4 种，在策略推理模型中有 1 种。

策略依赖模型模式下的 Actor Element：

：Actor 主体元件

：Agent 主体元件

：Role 主体元件

：Position 主体元件

策略推理模型模式下的 Actor Element：

：SRActor 主体元件

Intentional Element：

：目标依赖

：软目标依赖

：任务依赖

：资源依赖

Link：

：Decompositon

：Contribution

：Covers

：Dependency

：INS

：ISA

：Ispartof

：MeansEnds

：Occupies

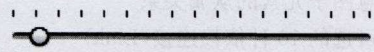
：Plays

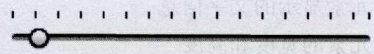
3. 属性栏

属性栏主要用来对选中的元件进行设置，包括元件的显示文字、元件的长度和宽度等。在策略推理模型的编辑模式中，属性栏还会增添为 SR 主体添加子元素及子元素解除从属关系的按钮。

策略依赖模型模式下的属性栏：

Text：Actor ：设置元件显示文字

Width：  ：设置元件宽度

Height：  ：设置元件高度

策略推理模型模式下属性栏增加部分：

Add：Add Child ：为 SRActor 元件增添子元件

Remove：Remove Child ：移除子元件的从属关系

4. 绘图面板

绘图面板用来显示及编辑模型图，绘图面板允许增添、选中、删除及拖拽元件和连接，同时整个面板也支持拖拽和放缩，给编辑过程带来了极大便利，大大地提升了用户体验性能。

2.2.2 代码结构

1. 客户端代码

客户端主要使用 Flex 技术进行开发，主要负责界面交互、图形绘制等功能，还使用 Flash Builder 4.6 工具开发除主程序文件外的 3 个包：component、logic 和 model。具体文件功能如下：

Lerisk. mxml：Flex 主程序文件，是整个界面的显示平台

Session. as：存储全局变量并配置资源

· component 包

GraphUnitView. mxml：意图元件及策略依赖模型中主体元件组件

LineView. mxml：直线/曲线连接组件

SRActorView. mxml：策略推理模型中主体元件组件

· logic 包

GraphUnitFactory. as：图素生成工厂类，根据输入图素信息生成组件

LinkRule. as：连接合规文件，用以检查连接合法性

· model 包

BitmapBook. as：图素信息与位图关系字典，用以查找相应位图

Config. as:图素配置文件

Data. as:模型图根数据结构,记录一幅图的所有图像信息

GraphInfo. as:图的 3 种结构列表信息

GraphUnit. as:意图元件及策略依赖模型图模型类

LineIconBook. as:连接类型与相应位置关系字典

LineModel. as:连接元件模型类

Point. as:坐标类

SRActor. as:策略推理模型主体元件类

除 src 代码文件包外,客户端还有 image 文件夹用于存放图片素材,libs 文件夹用于存放库文件等。

2. 服务器端代码

服务器端主要使用 Java 进行开发,主要负责云存储、布局计算、实例序列化等功能,还使用 Eclipse 进行开发,除 com. edu. tsinghua. lerisk 和 com. edu. tsinghua. lerisk. model 两个包之外,其它都为进行自然语言需求文本处理的代码实现。具体功能如下:

· com. edu. tsinghua. lerisk 包

ServiceProvider. java:主服务接口,用以接受并返回客户端的各种请求

Lerisk. java:主服务器逻辑类,实现各种服务端功能

Layout. java:自动布局类

GraphUnitType. java:图素定义类,与客户端对应

Config. java:总图素配置,与客户端对应

· com. edu. tsinghua. lerisk. model 包

Data. java:模型图根数据结构,与客户端对应

GraphInfo. java:包含模型图 3 个主数据列表,与客户端对应

GraphUnit. java:意图及策略依赖模型图元件类,与客户端对应

LineModel. java:连接类,与客户端对应

SRActor. java:策略推理模型主体类,与客户端对应

Point. java:坐标类

Vertex. java:用于自动布局计算的顶点类

另外还有一个 config. dat 的文件,即元件库配置的序列化存储文件,在服务器开启时被读入初始化元件库。在 flex 文件夹下存储着与 BlazeDS 相关的配置文件。

2.2.3 主要功能实现方法

1. 前后台通讯模块

通讯模块主要使用 Blaze Data Service 及 Remote Object 技术实现。在服务器端和客户端有几个名字和结构完全相同的类,该技术在通信时进行自定义类识别时会进行统一的操作。此外在服务器端定义了一个专门负责网络通信的类:ServiceProvider,使用远程对象的技术在 flex 文件夹里的 remoting-config 中的配置方法如下:

在〈destination/〉中添加 id,名为 ServiceProvider,同时在〈properties/〉添加子元素〈source/〉,把 ServiceProvider 类的完整包路径添加进去。之后在 flex 中定义好相对应的远程对象,就可以调用后台方法并接收返回函数。

2. 基本模型图编辑功能

整个模型图的数据结构由 GraphInfo 类定义实现,其主要域如下:

boolean isSDModel:策略依赖模型图/策略推理模型图

ArrayList〈GraphUnit〉 unitData:意图元件及 SD 图的主体元件列表

ArrayList〈SRActor〉 sractorData:SR 图主体元件列表

ArrayList〈LineModel〉 linesData:模型图连接列表

上述 3 个列表记载了模型图中所有元件的状态,每个列表的元素有相应的数据结构,包括元件的长度、宽度、显示文字、坐标等。在此基础上所有对图的编辑都是通过改变 GraphInfo 中各元件的状态实现的。

(1)新建模型图

点击按钮"New",弹出对话框,单选按钮组分别代表策略依赖模型图和策略推理模型图,点击"OK"将面板刷新开始所选模型图绘制,点击"Cancel"取消操作。

点击"OK"将会据所选按钮重置 isSDModel 的值,同时清空图形 3 个列表,切换工具面板 ActorElement 控件组。

(2)增添主体及意图元件

拖拽左侧选中的元件及面板,面板将会释放鼠标的位置,以默认值绘制相应的元件。

左侧控件栏使用的是 Flex 的 TileList 组件,在网页载入时,程序会通过 ServiceProvider 远程对象调用 loadData 的请求,将后台读入的元件库动态加载至本地的 TileList 组件中。拖拽的过程使用了 Flex 的 drag 效果系统,面板接收到拖拽物时根据被拖拽物的类型在 GraphUnit 中相应列表添加相应的实例,并根据新的数据进行面板重绘。

(3)增添连接

拖拽左侧选中的连接至画板,右上角提示"@Drawing Line Phase",表示当前处于画线阶段,点击两个可以连接的元件来完成画线。添加连接时工具会进行合法性检查,若连接对于相应元件不合法,则弹出窗口提示连接无法创建。

Boolean 域 drawLinePhase 表示当前是否处理画线模式,当它为 true 时,将对其后点击的两个元件进行连接创建,在 linesData 中加入新的元素并结合当前的直线/曲线连接模式进行绘制并进行合法性检测。连接的曲线绘制使用了二次贝塞尔曲线的技术。LinkRule 主要负责进行检测,Lerisk 对每种连接都实现了其检测规则,共 10 条,如表 1 所列。

表 1 Lerisk 连接合法性检查规则

连接名称	合法连接
Dependency	任意两个元件
Decomposition	意图元件指向 Task 元件
Contribution	任意元件指向 Softgoal 元件
MeansEnds	意图元件指向 Goal 元件
ISA	SD 主体元件指向 SD 主体元件
INS	SD 主体元件指向 SD 主体元件
Play	Agent 主体指向 Role 主体元件
Occupies	Agent 主体指向 Position 元件
Covers	Position 主体指向 Role 元件
Ispartof	SD 主体元件指向主体元件

(4)选中拖拽元件及画板放缩

鼠标左键单击可以选择元件及连接,并使选中的图素发出光晕;可用鼠标左键拖拽元件及整个画板,并通过鼠标滚轮向上及"+"按钮来放大画板,用鼠标滚轮向下及"-"按钮来缩小面板,用"1"按钮恢复原始大小。

元件内部有域 isSelected 来维护表示该控件是否被选,若被选,则使用 Flex 的 Glow 效果进行动画;同时整个面板也维护一个 selectedUI,用以维护当前被选择的控件。鼠标在元件上拖拽该元件,在空白区域时对整个画析进行拖拽,使用

Flex 的 Drag 效果，并将其坐标进行实时更新，当其有连接时，其连接也会在它移动时进行动态绘制。而画板的放大和缩小则是使用了 Flex 的 Zoom 效果。

(5)通过属性栏调整元件

在选中某元件后，属性栏的控件就会由不可用变为可用，在“Text”的文件框中输入文字再按 Enter 键就可以更改元件显示的文字，拖拽“Width”“Height”的滑杆可以调整相应的宽度和高度。在编辑策略推理模型图时，选中 SRActor 时属性栏会出现“Add Child”的按钮，点击该按钮，右上角蓝字会提示“Adding Child Phase”，再点击某元件则会在主体中加入该元素；选中意图元件时，属性栏会出现“Remove Child”的按钮，点击该按钮则会移除该意图元件的从属关系。值得一提的是，当意图元件与某主体有从属关系时，拖拽该主体将会带着所有子意图元件一起移动。

属性面板的 Enable 属性由当前模式决定，“Text”文本框直接改变元件数据结构实例的“Lable”，即显示文字属性，另两个滑杆也与其内部长度和宽度有直接联系。域 addChildPhase 表示当前是否处于添加子元素状态，意图元件结构类中有域 actor，记录它所属的主体。当主体被拖拽时，将所有 actor 值为它的意图元件同时进行拖拽，包括各元件的连接也进行即时绘制。

(6)删除元件及连接

选中某连接时，点击“X”将会删除该连接；选中某元件时点击将会删除该元件及所有与之相连的连接；在策略推理模型图中选中主体时，点击将会删除该主体及其所有从属意图元件及各元件所联系的连接。

删除连接时，在 linesData 中直接移除该元素并进行重绘；删除元件时，在 unitData 中删除该元件，并在 linesData 中查找所有与之相关的元素并删除；删除 SRActor 时，在 sractorData 中移除该元件，在 unitData 中查找所有其从属元件并移除，在 linesData 中移除所有与从属元件相关的元素。

(7)本地及云端存储

点击“Save”将弹出对话框，再点击“OK”将会弹出对话框选择本地保存路径，并成功保存至本地。点击“Load”将弹出对话框选择本地文件，选择保存的文件，Lerisk 将其加载并绘制在画板上。点击“Upload”将弹出对话框，输入文件名并确定，Lerisk 将文件存储在服务端。点击“DownLoad”并输入正确的文件名，程序将远程加载并绘制该模型图。

在服务器端使用了开源的 XML 序列化工具库：“XStream”，该工具可以将 Java 的实例序列化为 xml 文本，同时支持反序列化，由 xml 文本得到原实例。在本地保存时，远程调用 ServiceProvider 的 serialize 方法，将当前模型图信息的 GraphInfo 实例进行序列化，并返回 xml 文本，前台再将 xml 保存为本地文件。本地载入时读入 xml 文本再远程调用 unserialize 方法，将反序列化的 GraphInfo 实例返回给前台进行模型图绘制。云端存储的主要区别在于存储时将 xml 文本以指定名称存储在服务器上，所以实现了移动的便捷性。

3. 文本处理及自动布局功能

点击“Text”按钮在新弹出的对话框中输入需求文本或从文件载入，确定后 Lerisk 将会对需求文本进行建模，并将建模结果的模型图以随机布局的方式显示在画板上，以对比自动布局结果。此时点击“L”按钮便可进行自动布局。

请求处理文本时，远程调用 getXMLFromText 方法，将输入的文本送至服务端用文本处理模块进行建模，返回 xml 文件给前台。Flex 端再对 xml 文件进行解析，并用随机布局方式建立起一个 GraphInfo 实例，再进行画板绘制。在策略依赖模型图模式下请求自动布局时，远程调用 ServiceProvider的 layout 方法，将本地模型图的 GraphInfo 实例传至服务器端，新建 Layout 实例，并进行布局，将布局后的结果 GraphInfo 传至前台，再进行绘制。

4. 增添自定义元件功能

点击“NewIcon”按钮在新弹出的对话框中输入新元件名字，选择是 ActorElement 或 IntentionalElement，并浏览上传图标文件，确定后 Lerisk 将会在相应的控件栏里增添该元素，该元素可以拖拽至画板进行编辑，与相应其它元件具有相同的性质和功能。

在提交时，Lerisk 首先根据所输入的信息建立一个 GraphUnitType，然后远程调用 ServiceProvider 的 addGraphUnitType 方法将该实例提交至后台并加入到 config 实例中，最后实例化至 config. dat 文件中。服务器启动时将读入该文件，实例化所有的元件实例。

3 实验对比

3.1 工具部署方法

将提交的 Lerisk. war，直接放置在本机 tomcat 的 webapp 目录下，打开 tomcat 根目录，找到 conf/server. xml 文档，定位于〈Service name＝"Catalina"〉，向后寻找。找到 protocol＝"HTTP/1. 1"所在行。修改 HTTP/1. 1 协议的端口为 4000，浏览器访问网址：http://127. 0. 0. 1：4000/Lerisk/index. html。注意：tomcat 需要配置好 4 个变量：JAVA_HOME、CATALINA_HOME、PATH、CLASSPATH 以保证 Lerisk 代码中相对路径的访问。Lerisk 项目需要加载的文件 config. dat 在 Lerisk 根目录下。在 classes 文件夹下，有两个文件使用了相对路径：Lerisk. class 和 ServiceProvider. class。相对路径表达式采用类路径定位的方法，相应语句为：

```
String path=
new ServiceProvider(). serviceProviderPath;
//取出工程路径 Lerisk
path=
path. substring(1,path. indexOf("WEB-INF/classes"))+
"config. dat";
```

3.2 实验结果

3.2.1 需求文本处理及自动布局

打开 Lerisk 后，点击“Text”按扭，在文本框里输入以下文本，然后点击确定按钮。本实验对下列文本进行建模，并进行展示，如图 6 所示。

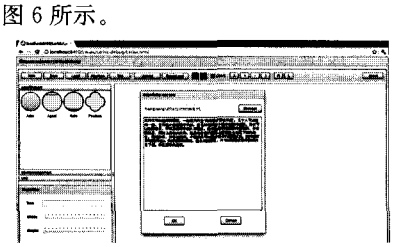


图 6 Lerisk 文本处理操作界面

输入文本：

本科生可以选修某课程，一起做大作业“完成项目管理系统”。3 个人可以组成一组。首先该小组注册本系统，组长被

老师赋予项目创建的权限。组长可以创建项目，并将其他同学加入本项目。组长负责制定项目的里程碑，并分配任务，同时上传相关文件。组员在自己的页面上查询组长分配下来的任务，完成任务，并标记完成。小组成员可以查看项目的进展状况，获得项目甘特图(可选)。最终项目到期时，组长关闭项目，并可以将项目所有资料打包下载。学生要求系统快。

随后 Lerisk 将会绘制其处理后的策略依赖模型，为了对比自动布局功能，默认先以随机布局形式进行绘制，其结果如图 7 所示。

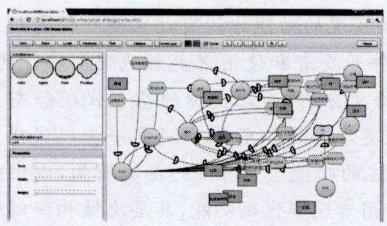


图 7　Lerisk 文本处理结果随机布局结果

Lerisk 提供对策略依赖模型图的编辑功能，如点“L”按扭来进行自动布局，Lerisk 将会进行布局并将结果重新绘制，其结果如图 8 所示。可见 Lerisk 不仅完成了自然需求文本的策略依赖模型建模，还完美地实现了其自动布局算法。

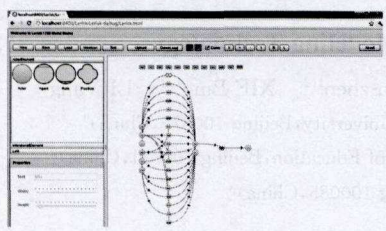


图 8　Lerisk 文本处理结果自动布局结果

3.2.2　策略依赖模型图的编辑

Lerisk 提供对策略依赖模型图的编辑功能，如图 9 为用 Lerisk 绘制的一幅策略依赖模型图。

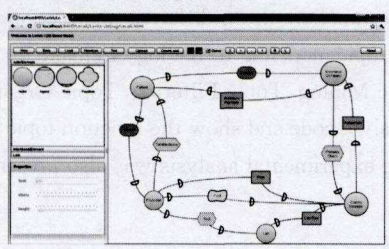


图 9　Lerisk 绘制策略依赖模型图

3.2.3　策略推理模型图的编辑

Lerisk 提供对策略推理模型图的编辑功能，如图 10 为用 Lerisk 绘制的一幅策略推理模型图。

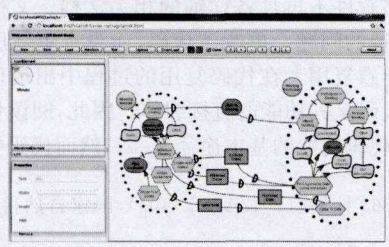


图 10　Lerisk 绘制策略推理模型图

3.2.4　扩展自定义图素

Lerisk 提供扩展自定义图素，点击“NewIcon”图标，在弹出的对话中填入 Label 信息：“Tom”，选择 ActorElement，并选择相应的图标图片，点击“OK”，如图 11 所示。

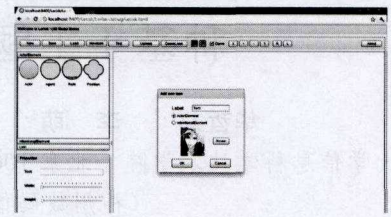


图 11　Lerisk 扩展自定义图素上传

之后可以发现该图标已经出现在 ActorElements 栏中了，可以将其拖出并绘制相应的模型图，如图 12 所示。

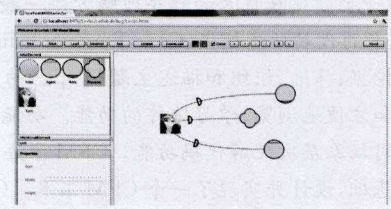


图 12　Lerisk 扩展自定义图素应用

3.3　功能对比

通过第 2 节工具的设计实现及第 3 节的实验结果，Lerisk 强大的功能已经可见一斑。下面将 Lerisk 与第 1 节中调研的 3 种主流 i* 建模框架工具功能进行对比，以展示 Lerisk 的功能特点，表 2 是 i* 建模框架基本功能对照。

表 2　Lerisk 与 3 种工具基本编辑功能对照表

功能	OME3	TAOM4E	Visio	Lerisk
创建	√	√	√	√
本地导入	√	√	√	√
添加元件	√	√	√	√
删除元件	√	√	√	√
直线连接	√	√	√	√
曲线连接	√	X	√	√
删除连接	√	√	√	√
重命名元件	√	√	√	√
调整元件大小	√	√	√	√
移动元件	√	√	√	√
本地存储	√	√	√	√

与此同时，我们也对比了它们的扩展功能。

表 3　Lerisk 与 3 种工具扩展功能对照表

功能	OME3	TAOM4E	Visio	Lerisk
连接合法性检查	√	√	X	√
扩展自定义元件	X	X	√	√
云存储与云导入	X	X	X	√
自动布局	X	X	X	√
画板拖拽与放缩	X	X	X	√
通用标准性	一般	较差	很差	良好

表 4　跨平台性、用户体验对比

软件	跨平台性	体验
OME3	仅支持 windows，跨平台性较差	较差
TAOM4E	插件，依赖 Eclipse，跨平台性很差	一般
Visio	依赖 Visio 软件，跨平台性一般	良好
Lerisk	依赖支持 Flash 浏览器，跨平台性好	良好

通过对比可以看出 Lerisk 不仅完整地实现了 i* 框架建模工具的基本功能，还提供了十分强大的扩展功能，同时在跨平台性、通用性和用户体验上也表现良好，是一个出色的 i* 框架建模工具。

(下转第 79 页)

中的3个组件，并用有向路径连接了这些组件。点击“运行”，分析流程提交到系统翻译执行，执行成功后的状态如图9所示。

大数据分析服务平台将数据分析应用开发变成了两阶段的任务。首先是编写自定义组件，提交到系统中，然后利用这些组件快速开发数据分析应用。系统集成了很多共享开发组件，用户只需要开发应用特定的组件即可，极大地提高了分析应用的开发速度。

结束语 本文为了解决大数据分析问题，设计并实现了一个可扩展的、分布式的、支持异构分析工具的、面向服务的大数据分析服务平台的原型。着重介绍了该平台的设计与实现，包括系统的整体架构以及流程映射、可扩展组件接口定义和中间数据的管理。在4.2小节中介绍了两种模型转换算法，方法A得到的是串行的执行序列，方法B对方法A进行了优化。但是由于JOIN的语义，每层的执行速度取决于执行时间最长的任务，因此方法B并不是最优的方法。进一步的优化方案将会在我们后续的工作中进行研究。

参考文献

[1] Islam M, Huang A K, Battisha M, et al. Oozie: towards a scalable workflow management system for Hadoop[C]// Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies (SWEET'12). 2012, 4: 1-10

[2] http://oozie.apache.org/docs/3.3.2/index.html

[3] http://www.cs.waikato.ac.nz/ml/weka/

[4] http://mahout.apache.org/

[5] http://www.r-project.org/

[6] 纪俊. 一种基于云计算的数据挖掘平台架构设计与实现[D]. 青岛：青岛大学，2009

[7] 余永红，向晓军，高阳，等. 面向服务的云数据挖掘引擎的研究[J]. 计算机科学与探索，2012，6(1)：46-57

[8] 钱肖鲁，朱建秋，朱扬勇. DMVisualMiner：一个可视化数据挖掘分析平台[J]. 计算机工程，2003，29：148-150

[9] 丁岩，杨庆平，钱煜明. 基于云计算的数据挖掘平台架构及其关键技术研究[J]. 中兴通讯技术，2013，19(1)：53-60

(上接第51页)

4 结论

4.1 总结

本文首先对主流的i* 建模工具进行了调研，研究了建模工具的基本功能，同时分析了其功能的不足，并在此基础上提出了新工具设计的功能补充点，给出了可视化工具的详细设计和实现。

i* 建模框架作为非常重要的需求工程分析框架，需要一个比较好的编辑工具。而通过我们的调研可以看到现在存在的工具不论是在功能上还是在用户体验性能上都趋于陈旧和粗糙。我们使用了富客户端技术的Flex进行开发，使得工具的用户体验大大提升，同时提供了云存储、自动布局等先进的功能，实现了一个较好的基于i* 框架的建模工具。

4.2 展望

虽然Lerisk已经是比较成型的一个i* 框架建模工具，但它仍然存在一些不足和缺陷，这些不足可以是对之完善的下一步目标。

1. 对于策略推理模型的自动布局功能

由于策略推理模型的主体元素和其它元件可能拥有从属关系，因此在进行布局时要考虑它们的包含关系。可以考虑将整个主体视作单位进行布局，再以对主体内部的元件进行布局的方式来进行实现，这可能是一个比较好的方向。

2. 更加完善的编辑功能

Lerisk虽然可以成功地实现整个流程的编辑功能，但在一些方面还有需要完善的空间，譬如元件的复制和粘贴、连接曲线的弯曲度调整等。

3. 性能的优化和提升

在处理的模型图非常庞大时，我们发现Lerisk效率会变慢，因此关于自动布局的算法及界面的绘制有进一步优化的空间。

希望这些尚未完善的问题能够在以后工作中得到解决，使Lerisk成为一个愈发成熟的建模工具。

参考文献

[1] 金芝，刘璘，金英. 软件需求工程：原理与方法[M]. 北京：科学出版社，2008

[2] Alonso O, Strogen J, Baeza-Yates R, et al. Temporal information retrieval: Challenges and opportunities[C]// Proceedings of the 1st International Temporal Web Analytics Workshop (TWAW) 2011. Hyderabad, India, 2011: 1-8

[3] Alonso O, Gertz M, Baeza-Yates R. On the value of temporal information in information retrieval[J]. ACM SIGIR Forum. ACM, 2007, 41(2): 35-41

[4] Schilder F, Habel C. From temporal expressions to temporal information: Semantic tagging of news messages[C]// Proceedings of the workshop on Temporal and spatial information processing-Volume 13. Association for Computational Linguistics, 2001: 9

[5] Pustejovsky J, Castano J M, Ingria R, et al. TimeML: Robust Specification of Event and Temporal Expressions in Text[J]. New directions in question answering, 2003(3): 28-34

[6] Dias G, Campos R, Jorge A. Future retrieval: What does the future talk about[C]// Proceedings of Workshop on Enriching Information Retrieval of the 34th ACM Annual SIGIR Conference, SIGIR. 2011

[7] Baeza-Yates R. Searching the future [C]// SIGIR Workshop MF/IR. 2005

[8] Kulkarni A, Teevan J, Svore K M, et al. Understanding temporal query dynamics[C]// Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011: 167-176

[9] Verhagen M. TimeML Corpora [DB/OL]. [2013-01-10]. http://www.timeml.org/site/publications/timeMLdocs/timeml_1.2.1.html

[10] Pustejovsky J. Task 15: TempEval Temporal Relation Identification [C/OL]. [2013-01-10]. http://timeml.org/tempeval/