# MarkTex项目文档

目录

[MarkTex项目文档 1](#_Toc375997376)

[需求 2](#_Toc375997377)

[Markdown介绍 2](#_Toc375997378)

[项目背景 2](#_Toc375997379)

[需求获取 2](#_Toc375997380)

[分析 3](#_Toc375997381)

[类图 4](#_Toc375997382)

[状态图 5](#_Toc375997383)

[构件图 5](#_Toc375997384)

[时序图 6](#_Toc375997385)

[协作图 7](#_Toc375997386)

[编辑器流程图 8](#_Toc375997387)

[转换器流程图 9](#_Toc375997388)

[显示模块流程图 10](#_Toc375997389)

[设计 11](#_Toc375997390)

[行为模式——命令(Command) 11](#_Toc375997391)

[行为模式——观察者(Observer) 11](#_Toc375997392)

[结构模式——代理(Proxy) 12](#_Toc375997393)

[实现 12](#_Toc375997394)

[开发环境 12](#_Toc375997395)

[运行方式 12](#_Toc375997396)

[附录 13](#_Toc375997397)

[项目链接 13](#_Toc375997398)

[组员工作 13](#_Toc375997399)

## 需求

本节先对Markdown作简要介绍，再对整个项目的需求获取过程进行说明。

### Markdown介绍

Markdown是一款轻量级的标记语言，使用它可以方便地将符合文法的纯文本文档转化为Html文档。它的特点是易读易写，相比Html注重文本表示的特性，Markdown则将重心放在了书写体验上。

### 项目背景

现在开源社区有许多版本的Markdown转换器实现，如Perl、Javascript和C#等，这为本项目提供了丰富的借鉴。此外，Window上还有一款易用但是收费的商用软件Markpad，这是本项目发起的重要原因。本项目的目标就是做出一款能够挑战Markpad的产品。

### 需求获取

为了达到上述的目标，我们将确定了系统要实现的主要功能：(1)通过图形用户界面进行编辑；(2)输入与输出同步显示；(3)支持将结果导出为Html或Pdf文档；(4)支持代码高亮和数学公式，以及其他编译器扩展。

此外，我们的转换器还实现了命令行接口(CLI)和Node包管理器(NPM)支持。

最后，我们抽象出了系统的用例图(图1)。

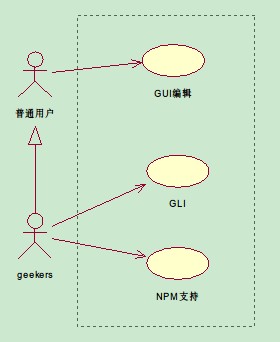


图1、用例图

## 分析

在系统分析阶段，我们运用了UML建模方法来提高系统的可扩展性。以下是本项目关键的UML图表(图2——图9)。

### 类图

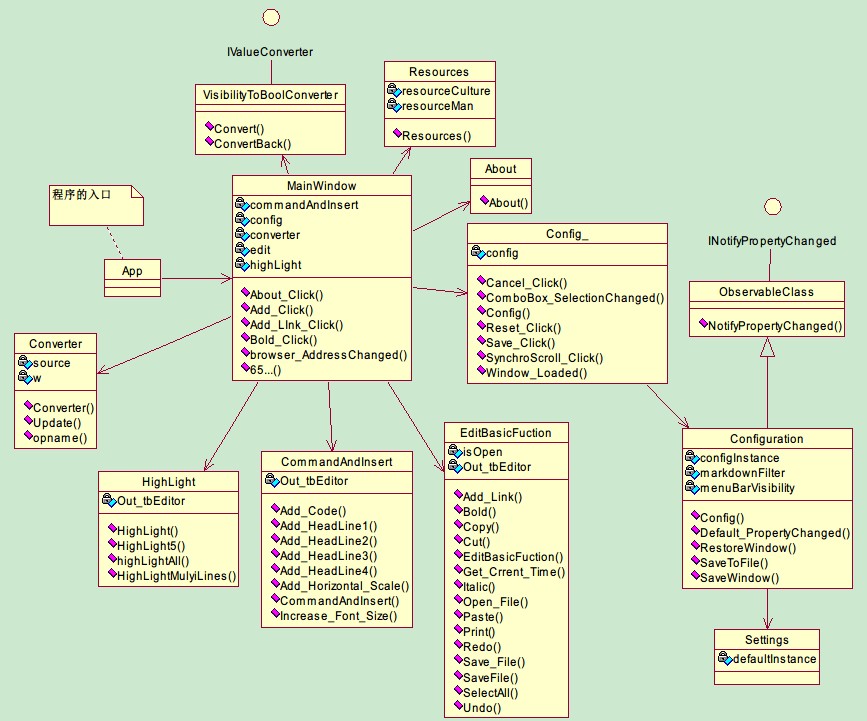


图2、类图

### 状态图

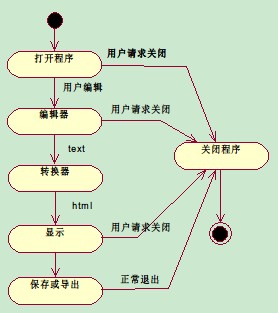


图3、状态图

### 构件图

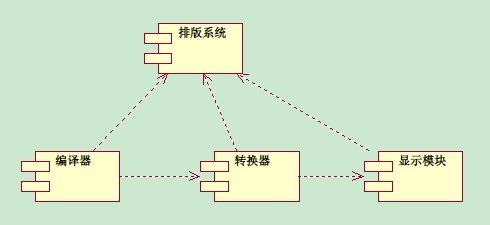


图4、构件图

### 时序图

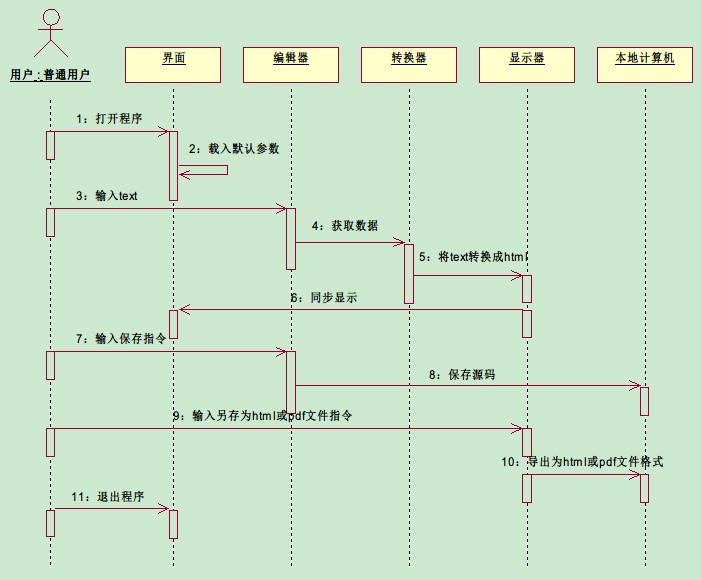


图5、时序图

### 协作图

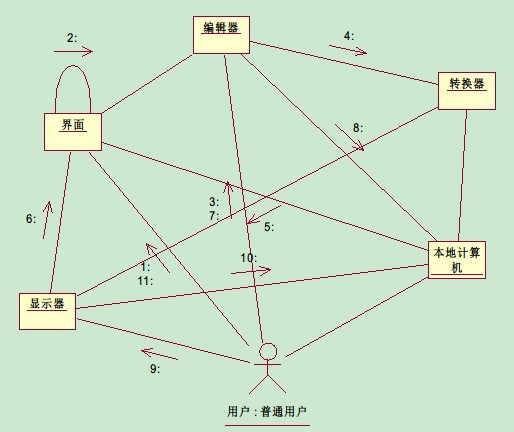


图6、协作图

### 编辑器流程图

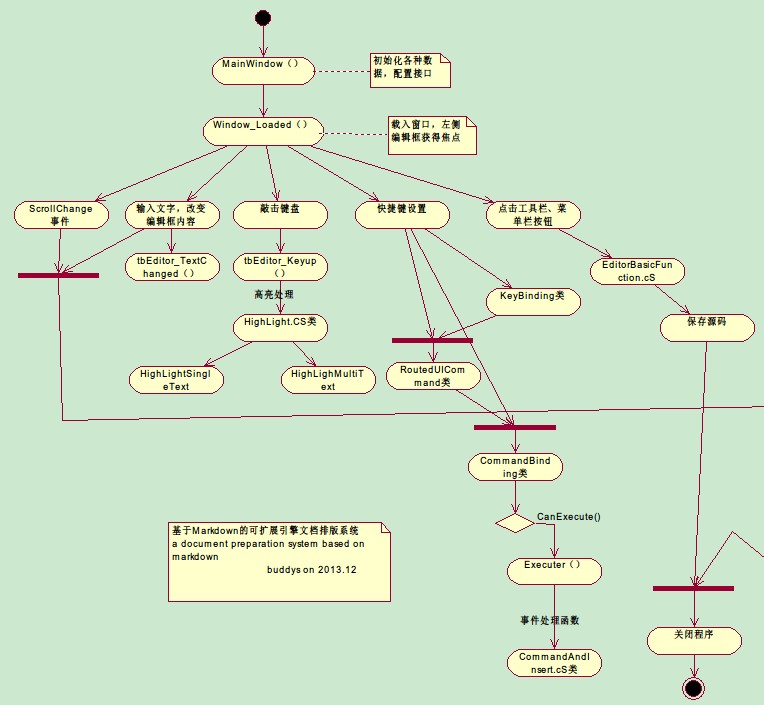


图7、编辑器流程图

### 转换器流程图

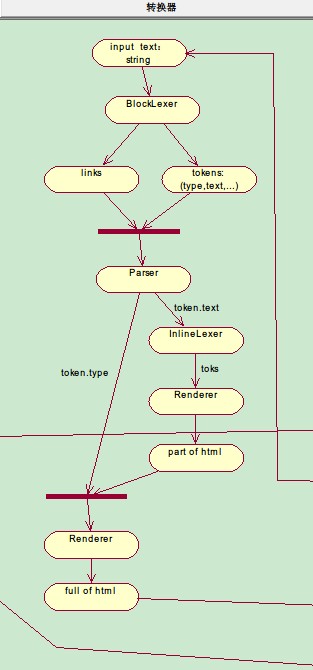


图8、转换器流程图

### 显示模块流程图

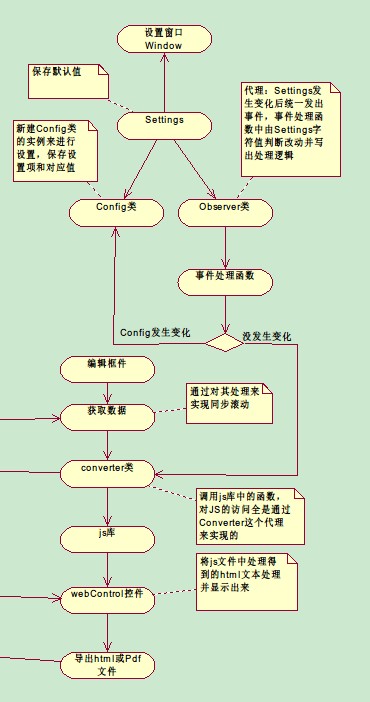


图9、显示模块流程图

## 设计

我们还运用了设计模式来解耦系统依赖和提高代码复用性。

### 行为模式——命令(Command)

Command机制将来自客户端的请求传入一个对象，无需了解这个请求激活的动作或有关接受这个请求的处理细节。该模式用于“行为请求者”与“行为实现者”解耦，可实现二者之间的松耦合，以便适应变化，分离变化与不变的因素。

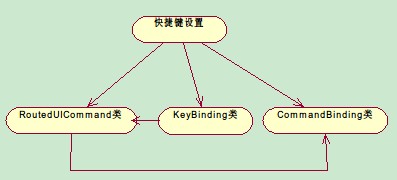
****

图10、命令模式示意图

### 行为模式——观察者(Observer)

OberservableClass作为观察者，检测Config变化。当Config发生变化时发出事件，由事件处理函数进行处理。

### 结构模式——代理(Proxy)

Converter.Source里定义了Set和Get方法来控制其他类中对这个对象的访问，向WebKit中传递数据必须通过这个代理(Proxy)。Set方法里定义了控制逻辑用来对字符串进行过滤，编辑器和显示模块的同步滚动的一部分就是通过代理来实现的。

## 实现

本项目托管于Github，采用Git版本控制系统进行协同开发。

### 开发环境

**MarkTexEdt**：

Microsoft Visual Studio 2010

Windows7 SP1

C#语言

WPF前台框架

**MarkTex**：

Linux 3.12.1-1-Arch

Javascript语言

NodeJS v0.10.22

GNU make 4.0

### 运行方式

MarkTexEdt：可通过ClickOnce安装包在windows下进行安装部署。系统要求：x86 CPU，.NET Framework 4.0

Marktex 是javascript框架，直接支持web开发；同时具有Unix命令行接口；同时提供npm模块，支持node系统。

## 附录

附录中包含了项目的Github托管地址和本组各自成员负责的相关工作。值得一提的是，在Github的项目首页当中有清晰易懂的项目介绍。

### 项目链接

**MarktexEdt**

项目主页：<http://github.com/buddys/MarkTexEdt>

发布页：<http://buddys.github.io/marktex>

User Guide：<http://buddys.github.io/marktex/guide.html>

Bug report：<https://github.com/buddys/marktexedt/issues>

**Marktex**

项目主页：<https://github.com/buddys/marktex>

发布页：<https://npmjs.org/package/marktex>

API DOCS：<https://github.com/buddys/marktex>

Bug report：<https://github.com/buddys/marktex/issues>

### 组员工作

杨珺 系统原型、编译器的设计与实现

杜实现 可视化编辑器的设计与实现

李淞毅 显示、设置模块的设计与实现

唐浩 系统分析与设计，答辩文档

张申 需求获取，报告文档

田璐 Logo设计，系统测试