iS3 需求分析规约

ID:GF-iS3

小 组 成 员

1751062 陈恬恬

1752461 刘雨

1754067 徐海琪

修订历史

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编写日期 | SEPG | 版本 | 说明 | 作者 | 评审时  间 | 参与人  员 | 批准日  期 | 确认人  员 |
| 2019-9-30 | iS3 | 1.0 | 初稿 | 陈恬恬  刘雨  徐海琪 |  |  |  |  |
| 2019-12-10 | iS3 | 2.0 | 二稿 | 陈恬恬  刘雨  徐海琪 |  |  |  |  |
| 2020-1-4 | iS3 | 3.0 | 定稿 | 陈恬恬  刘雨  徐海琪 |  |  |  |  |

目录

[修订历史 202](#_bookmark0)

1. [引言 5](#_bookmark1)
   1. [背景 5](#_bookmark2)
      1. [项目特色 5](#_bookmark3)
      2. [必要性 5](#_bookmark4)
   2. [参考资料 5](#_bookmark5)
   3. [假定和约束 5](#_bookmark6)
   4. [用户特点 5](#_bookmark7)
2. [运行环境规定 6](#_bookmark8)
   1. [设备 6](#_bookmark9)
   2. [与第三方平台接口 6](#_bookmark10)
   3. [普通接口 6](#_bookmark11)
3. [功能需求 7](#_bookmark12)
   1. [系统范围 7](#_bookmark13)
   2. [系统总体流程 7](#_bookmark14)
   3. [系统体系结构 9](#_bookmark15)
   4. [需求分析 11](#_bookmark16)
      1. [功能建模 11](#_bookmark17)
      2. [数据建模 27](#_bookmark18)
      3. [行为建模 29](#_bookmark19)

1 引言

1.1 平台背景

* + 1. 平台目标

iS3(infrastructure Smart Service System)平台的目标功能是：对地下工程从前期的地质勘测数据、地形数据、环境数据、设计数据、施工数据、监测数据一直到后期的运营维护等数据进行采集、管理、可视化和分析，服务于地下工程的整个生命周期。

* + 1. 平台特色

iS3 平台以开放性、集成性与全过程性为特点，可以服务与地下工程建设的各个阶段。在项目前期，iS3 可以根据所采集到的地质勘测数据对地层进行建模可视化，模拟出地下工程所处的地层情况。在设计阶段，iS3 可以通过掌1握的三维地层信息提出较为优化的选线方案，降低工程难度，节约工程造价；同时在地下结构设计方案确定以后，还可以将地下结构信息导出到专业有限元软件中进行分析；在施工阶段，可以利用 iS3 所采集的地质勘测信息和监测信息，指导现场施工，确保工程进度与安全，这将显著提高工程建设的管理水平。运营与维护阶段，基于 iS3 对于设计与建设过程中资料与信息的准确把握，可以根据维护数据对结构的健康状态进行快速评估，大大降低运营成本，提高维护效率，同时也可以提高地下结构灾害预防与快速处理的能力。

1.2 项目背景

iS3系统针对不同的用户群体及需求，推出了单机版和网络版。iS3单机版是适用于普通个人用户（如工程技术人员、科研人员、学生）的单机稳定版本，具备iS3平台的基本功能。

本次项目是针对iS3网络版C端进行开发，完成iS3 2.0 C端的一个工程案例部署，以及相应的软件部署文档、使用文档和二次开发接口文档，并在GitHub上挂接在线文档。项目的最终产出是一份关于iS3平台使用和开发的在线文档。

1.3 参考资料

[1]iS3平台使用及数据准备手册，同济iS3团队

[2]iS3平台开发帮助文档，同济iS3团队

[3]iS3系统架构介绍，同济iS3团队

1.4 假定和约束

**发布时间**：2019年年底完成iS3 2.0 C端稳定版本的发布，包括程序和软件文档

**使用条件：**iS3 主程序使用C#和Python语言，并基于.Net 4.5 Framework 框架开发而成。因此建议采用64 位 windows 操作系统，且系统至少为 Windows 7 SP1，处理器至少 1.6 GHz ，4 GB RAM， 并具备 11 GB 可用硬盘空间。

**经费支持：**无经费支持

**开发效率：**采⽤增量+迭代模型，产物可以分批次地进行交付，且降低了在一个增量上的开支风险，能有效地加快整个开发工作的进度。

1.5 用户特点

本项目面向人群如下：

1. 使用iS3平台的工程技术人员、科研人员、学生等普通用户，通过该在线文档可以了解如何使用iS3平台的基本功能；
2. 维护iS3平台使用的开发人员，通过该在线文档可以快速了解iS3架构和设计，便于开展维护工作；
3. 拓展iS3功能的二次开发人员，通过该在线文档可以快速了解如何通过C#和Python语言对原有平台功能进行拓展。

2. 功能需求

2.1 项目范围

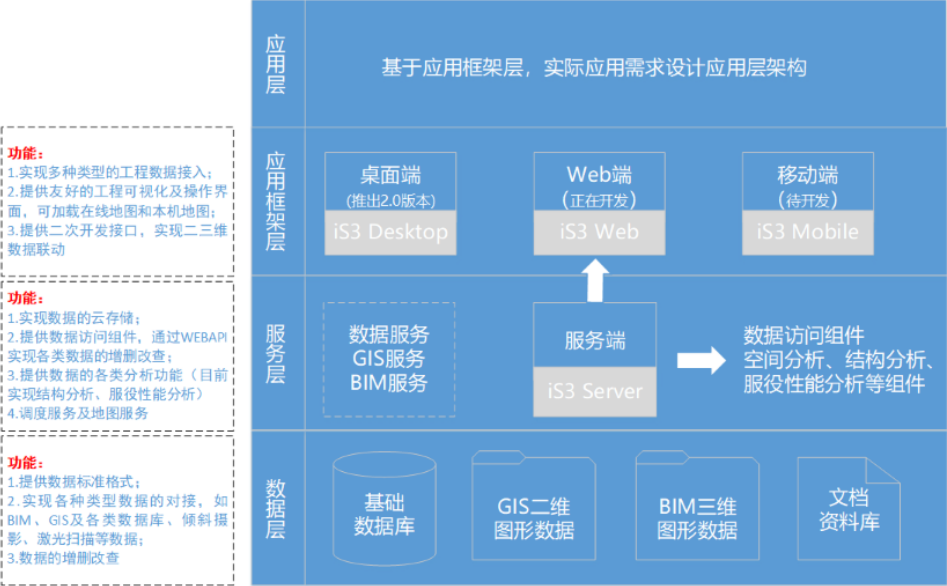
本次我们小组的研究，是在了解并能够使用iS3系统的前提下，产出一份用户的帮助手册，实现：

1. 普通用户可以根据这份手册，了解平台的功能；
2. 开发人员可以根据这份手册对系统进行维护；
3. 二次开发人员可以根据这份手册对iS3进行二次开发。

2.2 平台总体架构

系统总体目标：实现地下基础设施全寿命周期的数据采集、处理、表达、分析及决策一体化服务系统。

平台总体架构图如下：



2.3 项目总体流程

本项目拥有三个主要的业务流程，主要介绍iS3工程项目管理、iS3工程项目配置以及iS3的二次开发等，分以下三个业务流程图进行展示说明。

* + 1. 工程项目管理业务



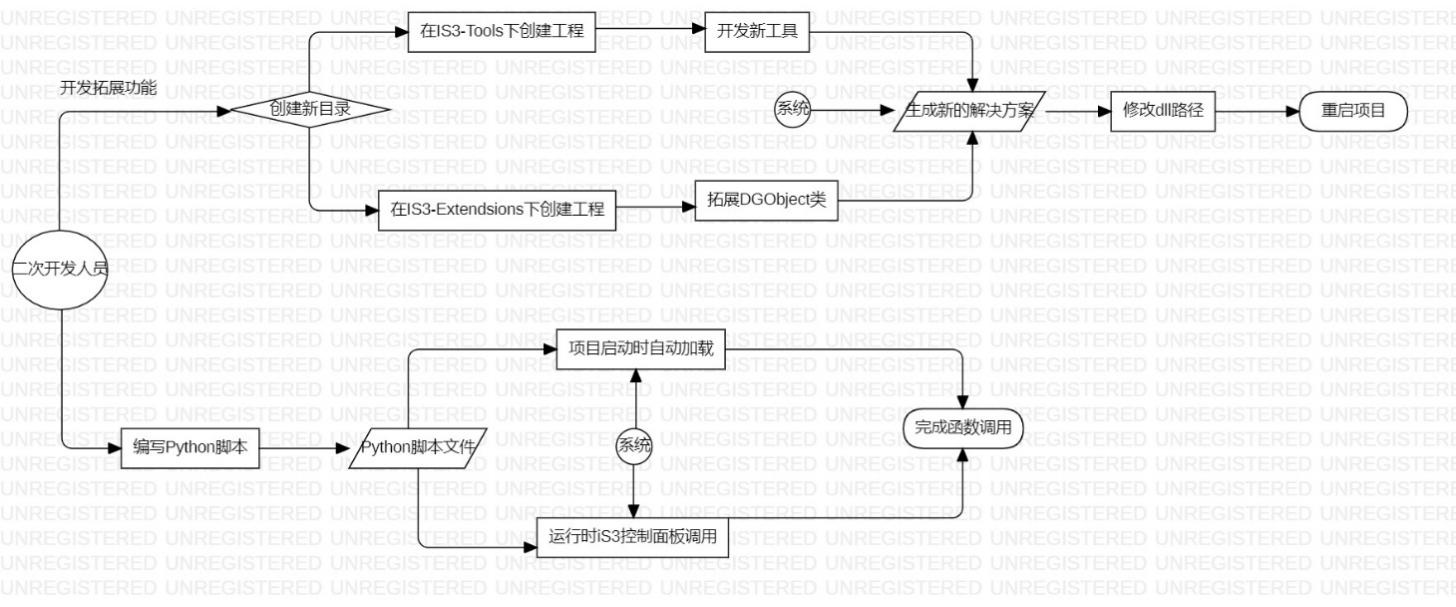
该项目是面向用户使用的，在登录后，系统会给出数据库中所有的项目信息列表，用户可以选择某个项目进行查看。在用户选择后，系统对该项目进行加载，加载后将项目的图形数据渲染出来并将工程数据展示出来，用户可以通过点击来查看图形数据与工程数据的关联。查看结束后，用户可以直接关闭窗口来结束程序。

* + 1. 工程项目配置业务



二次开发人员可以配置一个新的项目或修改一个项目的配置，这需要进行两步准备操作：配置iS3及数据目录的路径和配置项目地点和表述信息，四步操作：配置基本信息、配置2D模型、配置项目的Domains信息、配置项目树，并将最后配置完成的项目写入xml文件。

* + 1. 平台二次开发业务



本项目在该部分向二次开发人员介绍如何对原有的iS3平台功能进行拓展，可拓展内容包括：开发iS3平台的工具面板，自定义数据化对象类DGObject，以及编写可供用户调用的Python脚本函数。其中：开发工具面板需在IS3-Tools目录下创建工程，拓展数字化对象类需在IS3-Extensions目录下创建工程。Python脚本文件需要根据具体功能选择恰当的调用方式。

2.4 需求分析

* + 1. 功能建模

工程项目管理模块：

1. Login

描述对象:登录界面

活动图



说明：运行程序后，进入登录界面。如果使用默认IP和端口的Server端，使用默认账号和密码登录，直接点击确认，进行身份验证。通过验证后将跳转到工程列表页面。

1. ProjectListPage

描述对象：工程选择界面

活动图：



说明：进入工程选择界面后，从Server端获取工程列表信息，并将其加载到地图和侧边栏。当用户选择某个工程时，需要右键——进入该工程，此时页面跳转到主页面。

1. MianFrame

描述对象：主页面

活动图：



说明：进入主页面后，程序分别要加载Project和View，加载Project时，要分别加载定义信息、域信息和数字对象组信息。加载View时，要将图形数据文件的信息加载到每个view，然后将加载后的view放到视图列表中。完成后，需要将Project里的工程数据对象与View里的图形数据一一绑定，然后渲染到界面上。

工程项目配置模块：

1. 配置iS3及数据目录的路径

描述对象：准备步骤一界面

活动图：



说明：配置iS3的运行路径和数据目录的路径。默认情况下不用更改。

1. 配置项目地点和表述信息

描述：准备步骤二界面

活动图：



说明：先从文件ProjectList加载项目列表，加载地图、图标和加载出来的项目信息渲染在界面上。用户点击添加项目按钮，新建一个可以填写项目名和项目描述的窗口。用户填写完毕并确认后，可以在界面地图上选择一个点作为项目位置。完成后用户可以修改位置和描述信息，也可以直接点击下一步，保存项目信息后窗口关闭。

1. 常规配置

描述对象：步骤一界面

活动图：



说明：先从可能存在的项目文件中加载项目定义，将默认的数据文件路径显示在界面上。用户可以修改网格地图文件TPK的路径、地质图形数据文件geodatabase的路径，并填写存储数字对象对应地质数据的数据库的IP地址、数据库名称、用户名和密码。完成后，用户点击确认，关闭窗口。

1. 地图引擎配置

描述对象：步骤二界面

活动图：

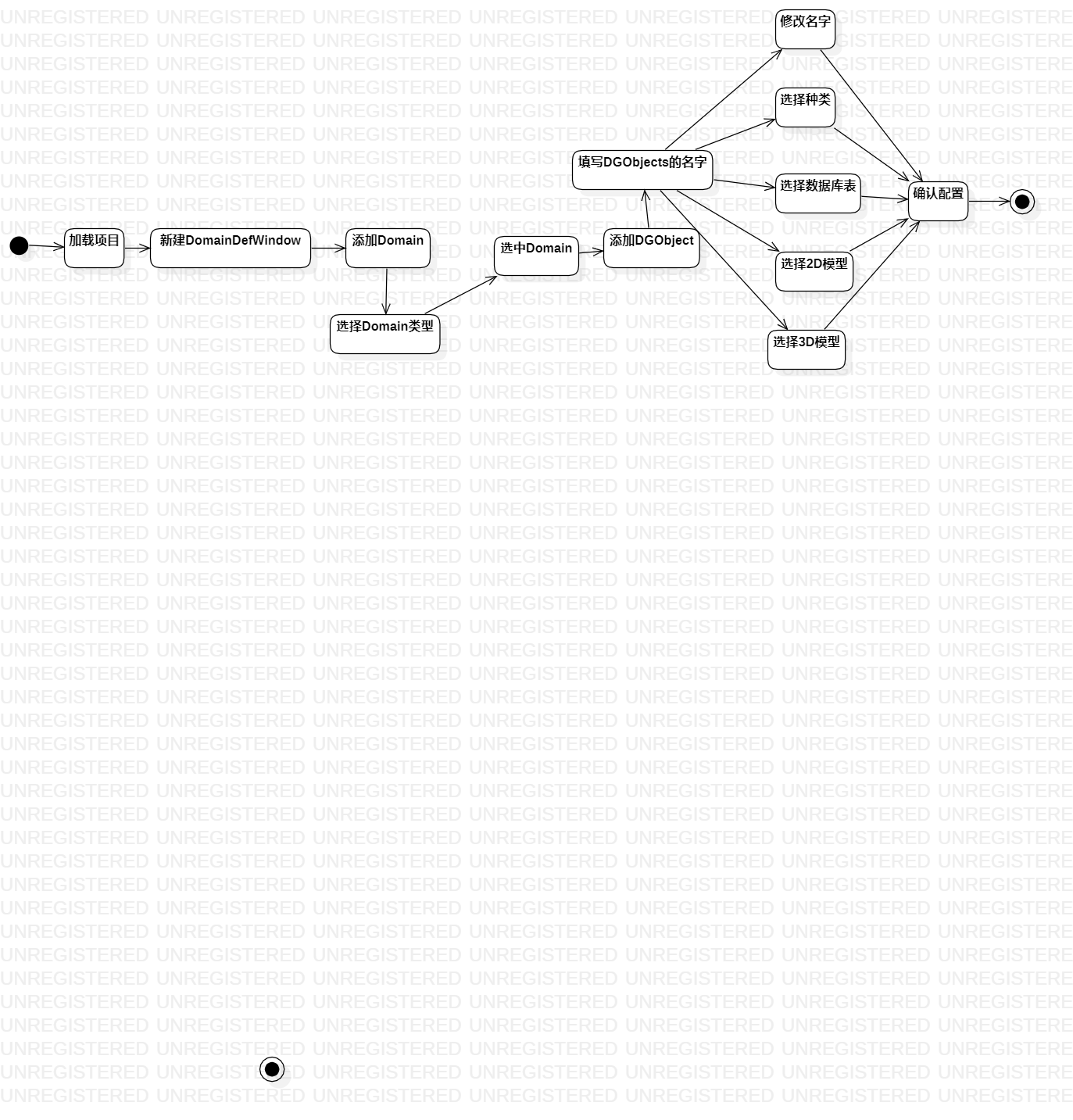


说明：先从可能存在的项目配置文件中将在可能存在的地图引擎并渲染默认的地图在界面上。当用户添加新的地图时先将地图添加到项目中，再添加到列表中。当用户选择某个地图时，可以进行修改地图的名称、种类、导入tpk文件和geodatabase文件四个操作，导入文件后在界面上渲染。完成之后，用户点击Next按钮。

1. 配置项目的Domains信息

描述对象：步骤三界面

活动图：



说明：从可能存在的项目配置文件中加载项目，将项目的Domain和DGObjects显示在界面上。用户选择添加Domain并选择Domain类型，即添加Domain成功。用户选中某个Domain，添加DGObjects，填写DGObjects的名字后，即添加DGObjects成功。用户可以对该数字对象数据组DGObjects进行修改名字、选择种类、选择数据库表、选择2D模型和3D模型等一系列操作。完成这些设定后，用户确认配置。

1. 配置项目树

描述对象：步骤四界面

活动图：



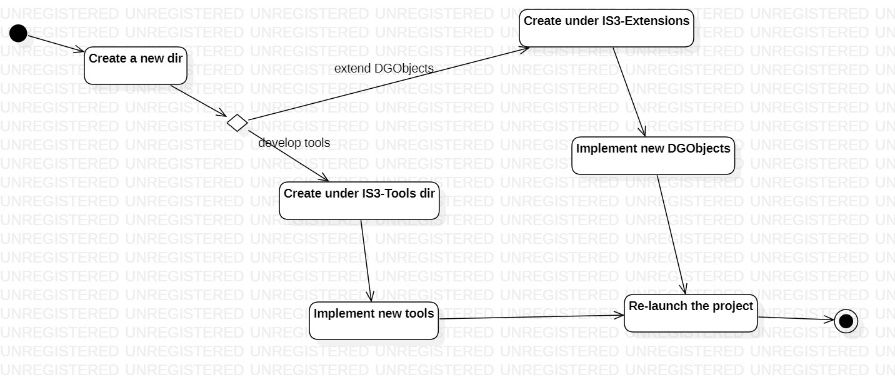
说明：用户新建树，填写树的名称。在树上添加节点，填写树的名称、节点显示名和对应的数字对象组。

平台二次开发模块：

1. Extend iS3 Project：

描述对象：了解扩展iS3平台功能

活动图：

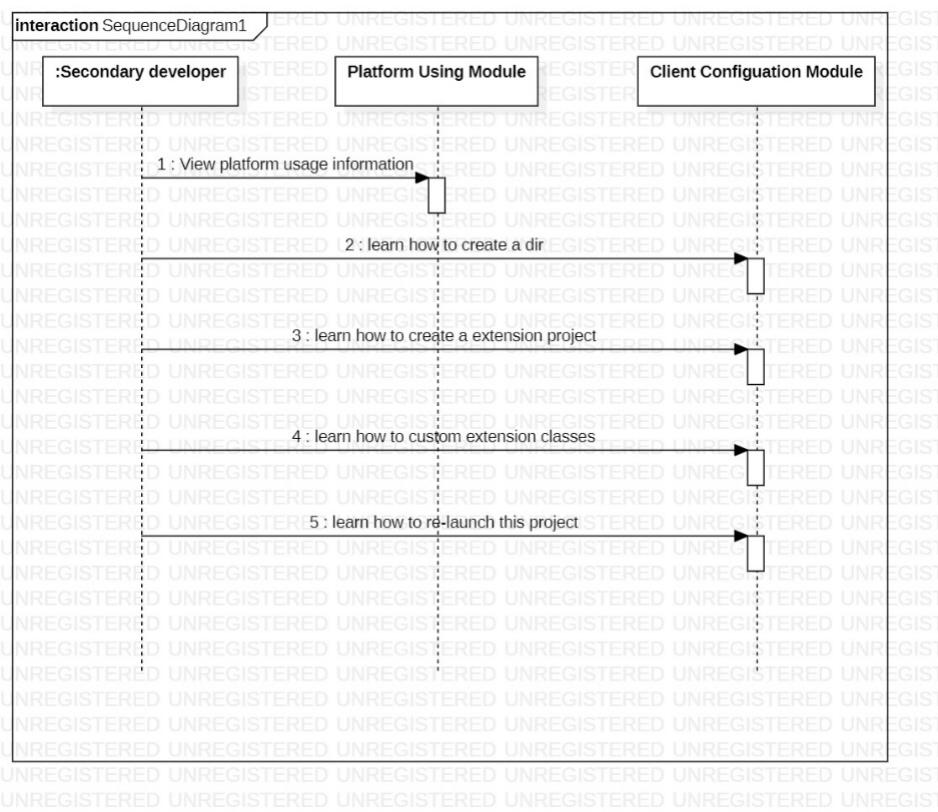


说明：本用例将向二次开发人员介绍如何在原有的iS3平台的基础上进行二次开发，增加新的功能点。对iS3平台的二次开发包括在工具面板添加新的工具，扩展iS3数字化对象DGOject类等。

基本操作流程：

1. 了解如何创建放置扩展程序的项目，包括：
2. 开发工具面板所放置的目录；
3. 拓展DGObject类所放置的目录；
4. 了解如何创建拓展项目；
5. 了解如何自定义有关拓展类，包括：
6. 开发工具面板的有关拓展类；
7. 拓展DGObject的有关拓展类；
8. 了解扩展后项目的启动方式。

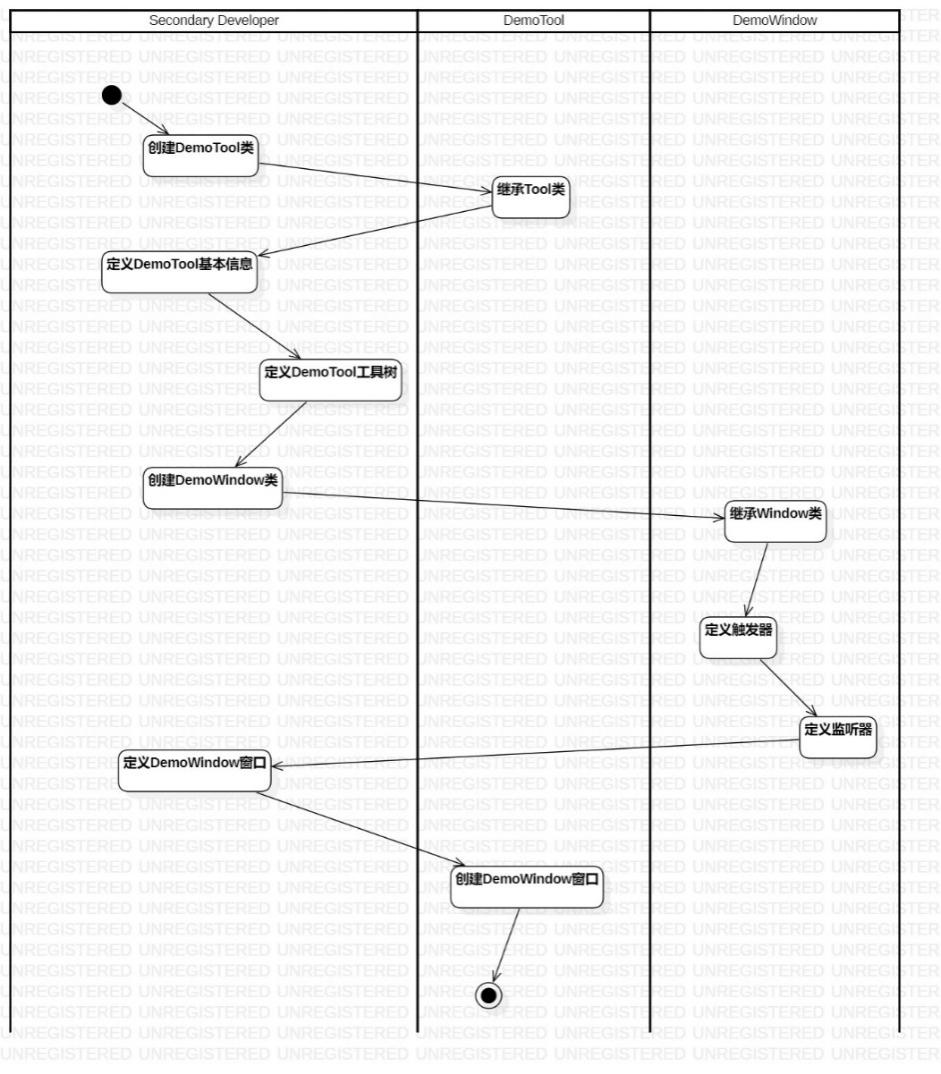
序列图如下所示：



1. Develop Tool：

描述对象：拓展iS3工具面板

活动图：

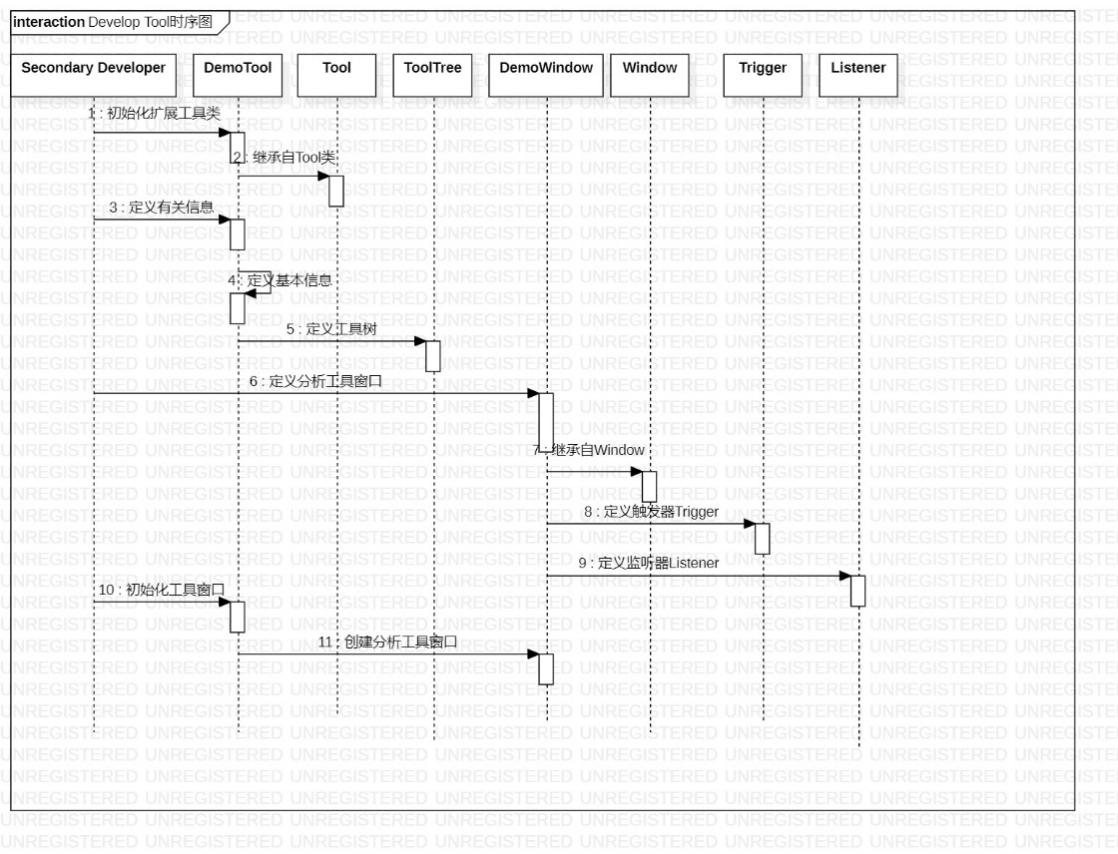


说明：本用例将向二次开发人员介绍如何对iS3的原生工具面板进行拓展，为其他用户提供新的使用工具。

基本操作流程：

1. 创建一个继承自Tools类的拓展工具类（DemoTool）；
2. 自定义DemoTool类的基本信息；
3. 定义DemoTool的工具树；
4. 创建一个继承自Window类的视图类（DemoWindow）；
5. 自定义DemoWindow的基本信息；
6. 定义DemoWindow的触发器和监听器；
7. 在DemoTool类中创建一个DemoWindow类的属性。

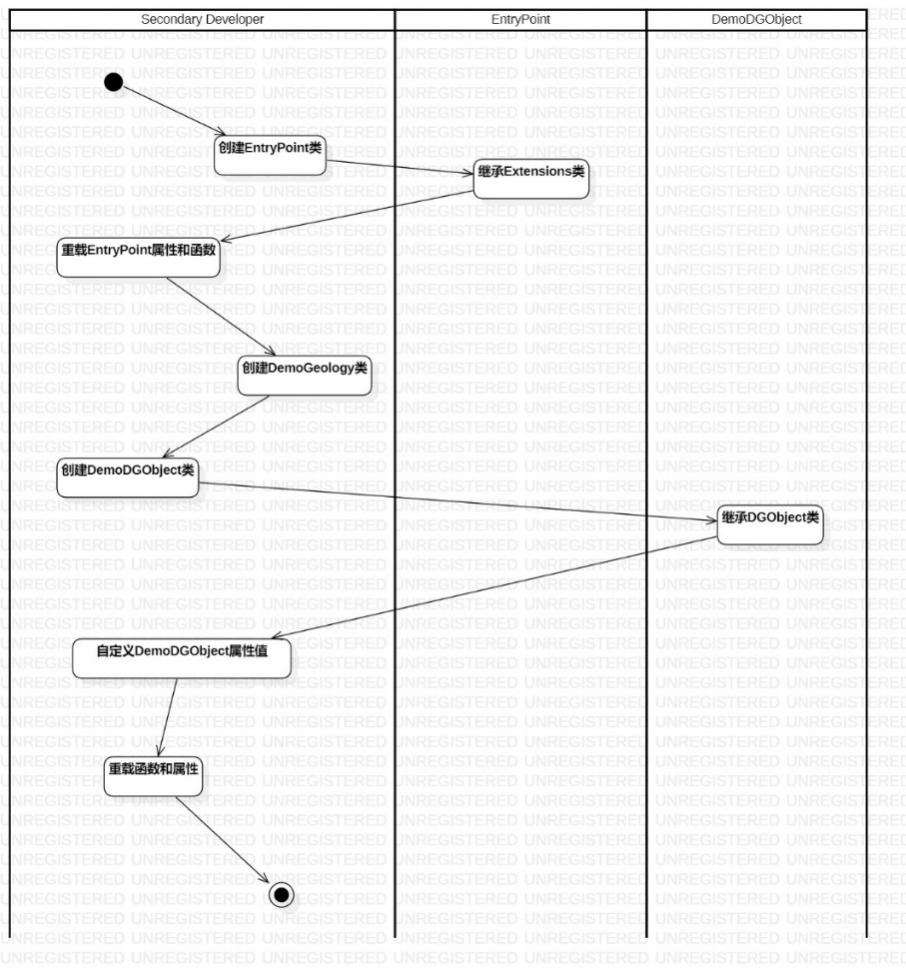
序列图如下所示：



1. Extend DGObject：

描述对象：拓展iS3的数字化对象类

活动图：

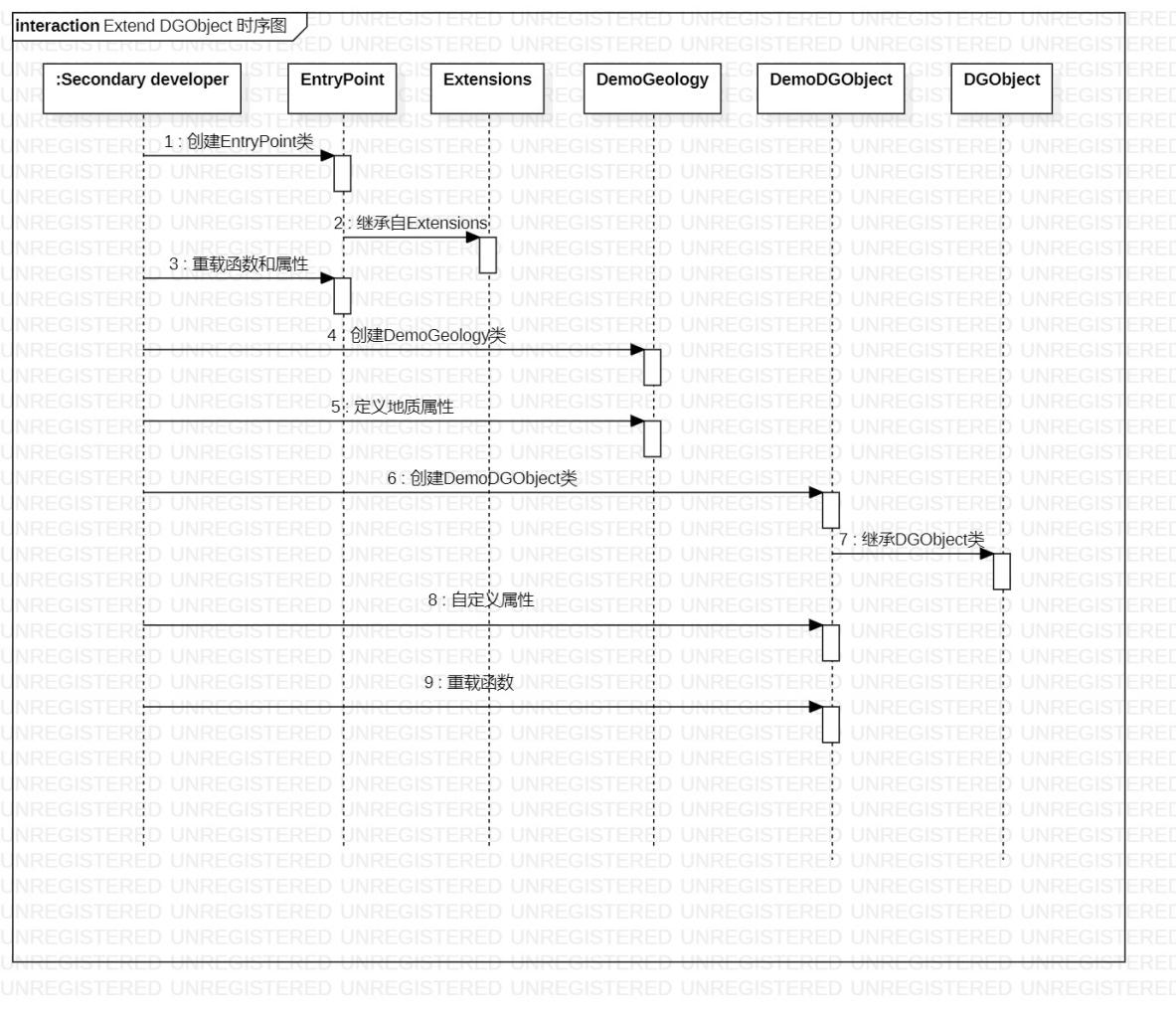


说明：本用例将向二次开发人员介绍如何在iS3平台已有的数字化对象（如Borehole类）的基础上，扩展新的数字化对象类，使得iS3平台能够呈现更多的工程对象信息，满足业务场景的需求。

基本操作流程：

1. 创建一个继承自Extensions类的扩展程序的入口类（EntryPoint）；
2. 重载入口类的属性和函数；
3. 创建一个用于渲染数字化对象的类（DemoGeology）；
4. 创建一个继承自DGObject类的扩展数字化对象类（DemoDGObject）；
5. 自定义该扩展数字化对象类的属性；
6. 重载该扩展数字化对象类的属性和函数。

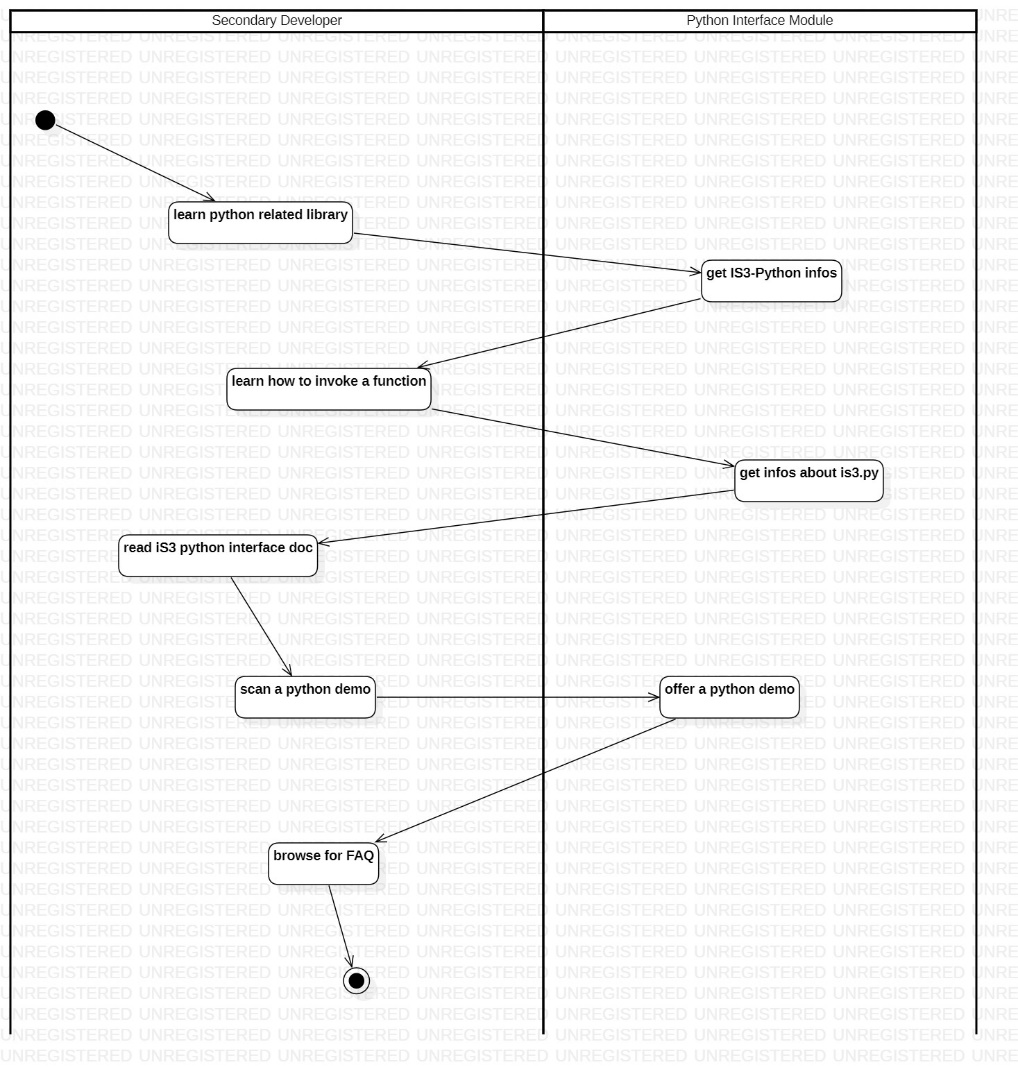
序列图如下所示：



1. Write Python Scripts：

描述对象：编写可供用户调用的Python脚本

活动图：



说明：本用例将向二次开发人员介绍如何使用Python语言进行二次开发，为其他用户提供在项目运行前和运行期间调用的脚本库。

基本操作流程：

1. 了解iS3工程项目的相关Python脚本库，包括：
2. IS3-Python工程；
3. PyPlugins扩展插件库；
4. IS3Py脚本库；
5. Output/Data目录下的Python配置文件。
6. 了解iS3平台上调用Python的方式，包括：

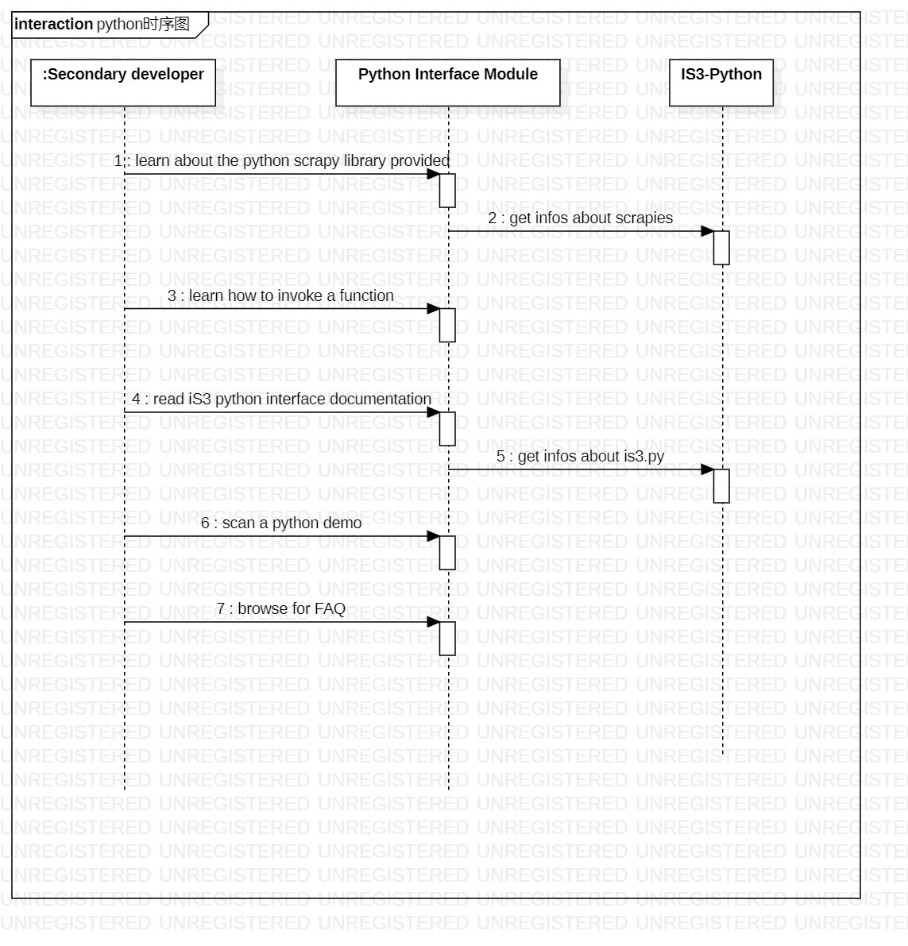
IronPython Console控制面板；

IronPython Pad脚本加载面板；

PyPlugins自动调用库；

1. 阅读iS3提供的脚本库接口文档；
2. 浏览iS3提供的Python脚本函数样例；
3. 阅读iS3提供的常见问题解答。

序列图如下所示：

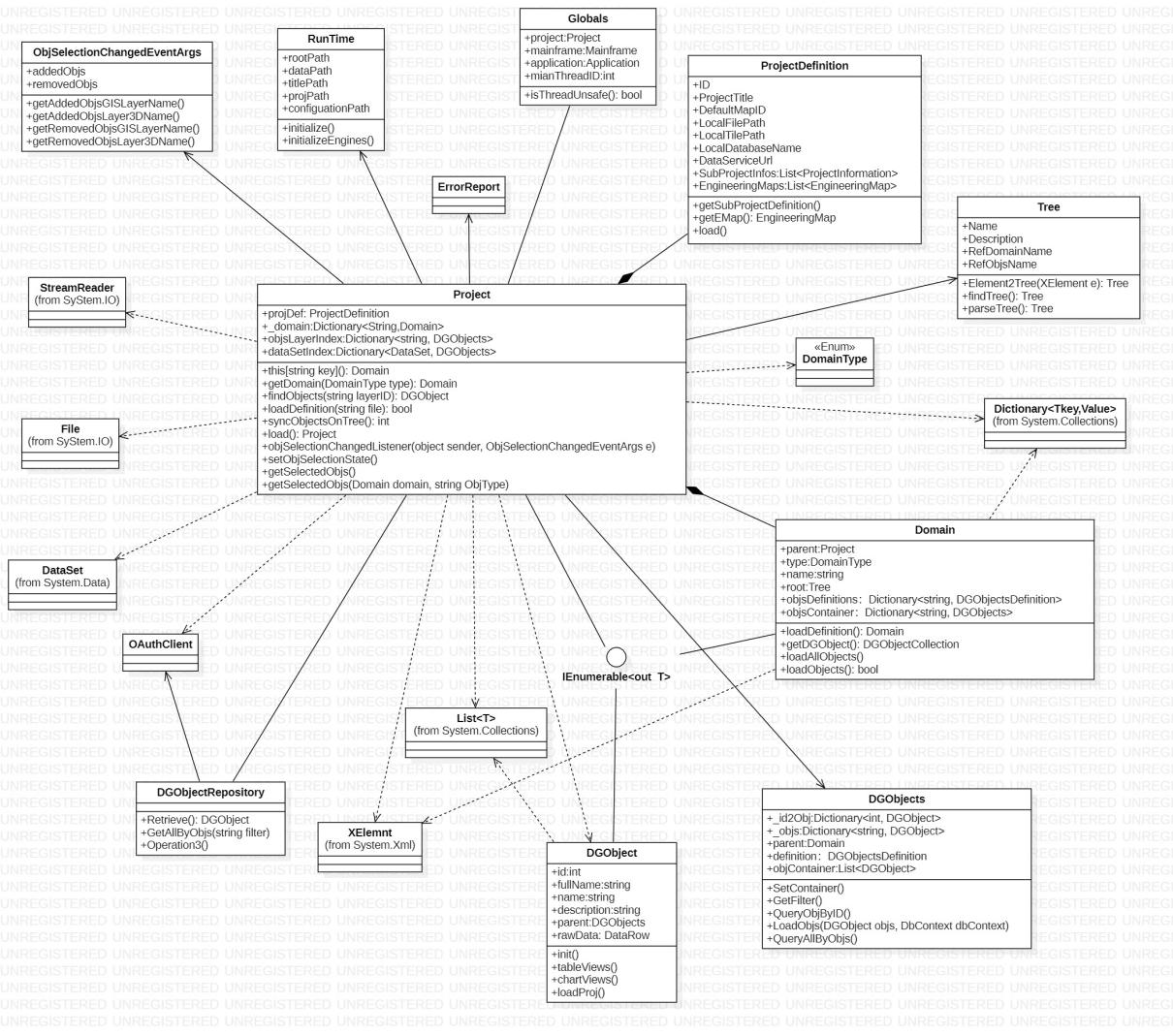


* + 1. 系统建模

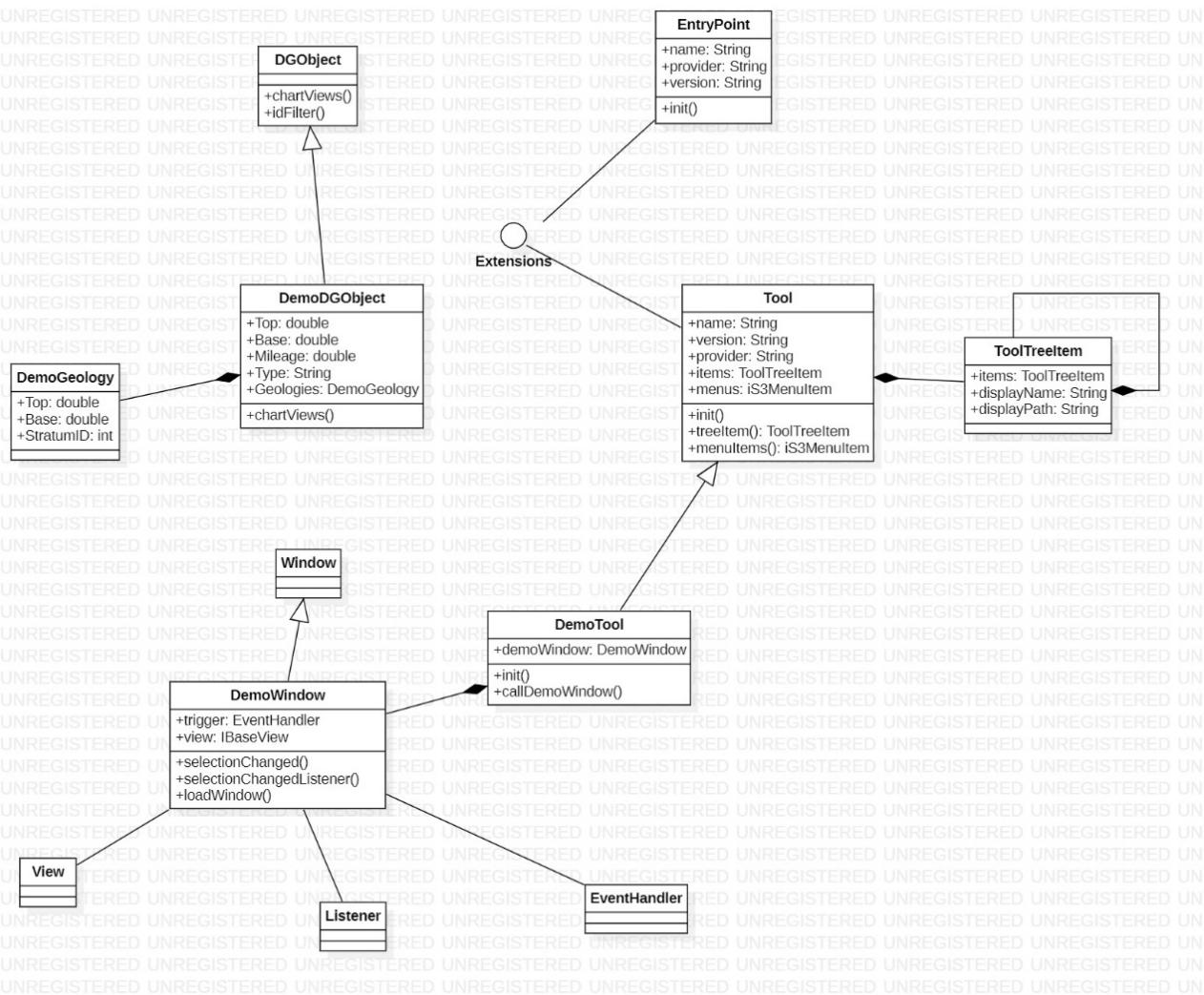
2.4.2.1 分析类图

此处分成两大模块展示分析类图。

工程项目管理及配置模块：



平台二次开发模块：



2.4.2.1 分析类

（类图中出现的分析类）

描述：类名，编号，职责，属性，说明

Project类

编号：  
职责：存储所有非图形数据

属性：项目定义projDef，域的集合domains

说明：Project类是MVC架构中的Model，是iS3平台核心类之一，从数据库中加载数据。Is3中只能有一个活跃的Project。

ProjectDefinition

编号：

职责：存储项目的一些定义信息

属性：项目子信息集合SubProjectInfos，地图引擎集合EngineeringMaps

说明：ProjectDefiniton类存储项目的定义信息，如图形文件存储在哪里，数据库的相关信息以及该项目共有多少域。

Domain类

编号

职责：存储工程域数据

属性：父项目parent，域的种类type，域名name，域下属的树root和数字对象的集合objsContainer

说明：Domain是一个工程概念，有地质geology，结构structure，监听monitoring等。数字对象被存储在objsContianer中。

DGObject类

编号：

职责：数字对象的基类。

属性：父数字对象组parent，编号ID，名称name，全名FullName，描述信息Description，

说明：DGObject被存储在DGObjects里，它是一个基类，可以被继承成为实体。

Extensions类

编号：

职责：识别用户自定义的拓展类，确保拓展类能被iS3平台识别并使用。

属性：命名空间name，当前版本version，提供者provider

说明：Extensions类是用户实现自定义扩展类所依赖的重要对象。所有用户自定义的拓展类均需继承自该Extensions类，否则无法拓展。

Tool类

编号：

职责：识别用户自定义的拓展工具类，确保拓展工具类能被iS3平台识别并使用。

属性：命名空间name，当前版本version，提供者provider，工具树列表items，菜单栏列表menus

说明：该类继承自Extensions类，用于实现用户自定义工具的拓展。用户自定义的工具均需继承自该类。

ToolTreeItem类

编号：

职责：在工具面板展示用户自定义的工具。

属性：子工具树items，展示名称displayName,展示工具displayPath

说明：该类用于展示用户自定义实现的工具类。DG

DemoTool 类

编号：

职责：实现用户自定义的功能。

属性：工具视图demoWindow

说明：该类是用户自定义实现的工具类。

DemoWindow类

编号：

职责：展示自定义工具的视图

属性：触发器trigger，视图view

说明：该类用于监听二次开发人员设置的触发事件，并在触发时加载自定义的视图信息。

DGObject类

编号：

职责：存储工程对象的基本信息。

属性：唯一标识符id，名称name，全称FullName，描述description，原始数据rawData

说明：该类将实际的工程对象抽象成iS3平台上的一个实体，后续对工程对象信息的获取和可视化均依赖于这个实体。

DemoDGObject类

编号：

职责：存储拓展的工程对象的基本信息。

属性：由用户自定义

说明：该类继承自DGObject类，是二次开发人员拓展的数字化对象类，以满足用户展示更多工程对象信息的需求。

DemoGeology类

编号：

职责：定义DemoDGObject的地质对象。

属性：由用户自定义

说明：该类定义了DemoDGObject的地质对象，目的是实现图表视窗对拓展的数字化对象DemoDGObject的图形渲染。

* + 1. 行为建模
       1. 项目工程管理模块

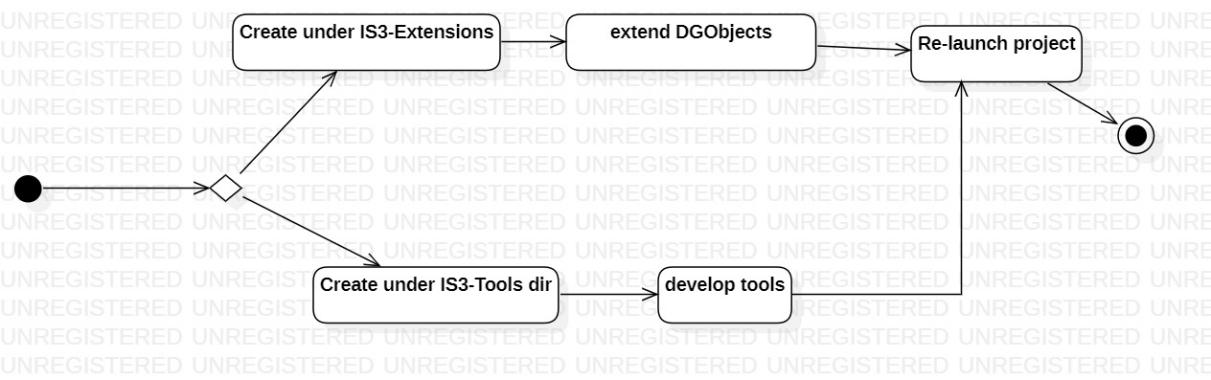


* + - 1. 项目工程配置模块



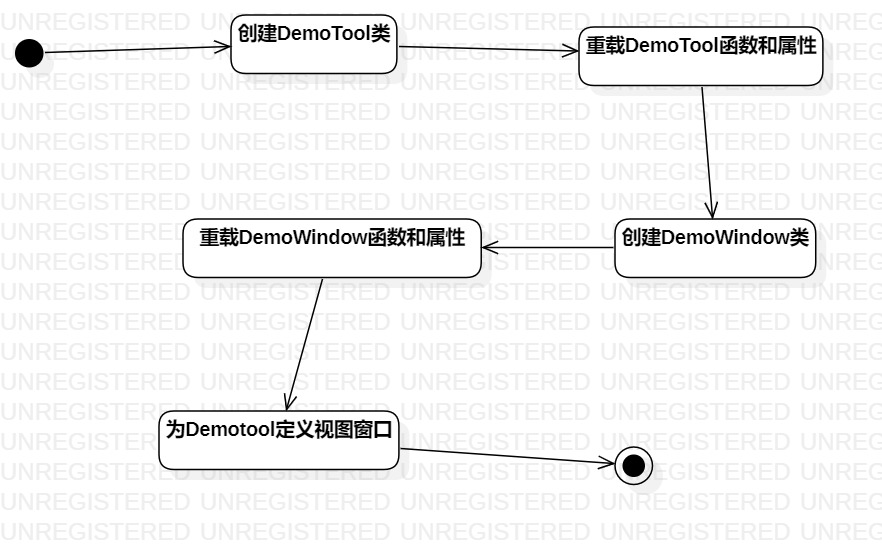
* + - 1. 平台二次开发模块

（1）Extend Project



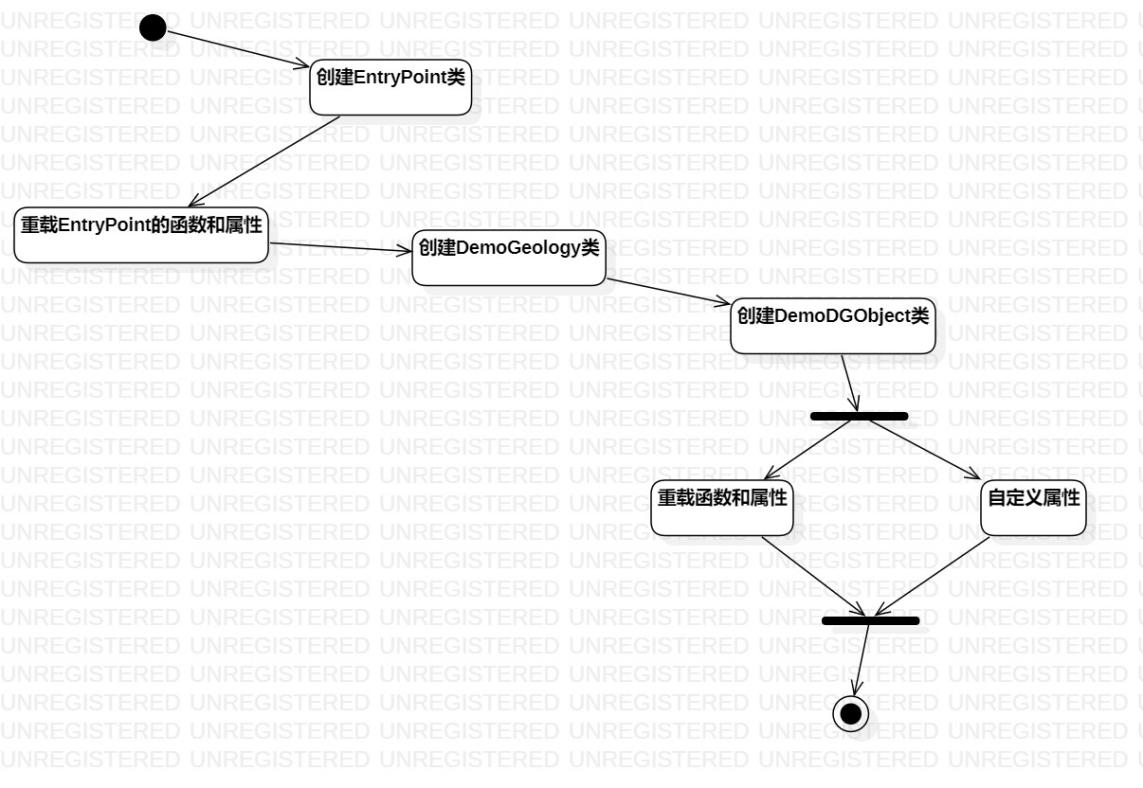
描述：对工程项目进行拓展。

（2）Develop Tool



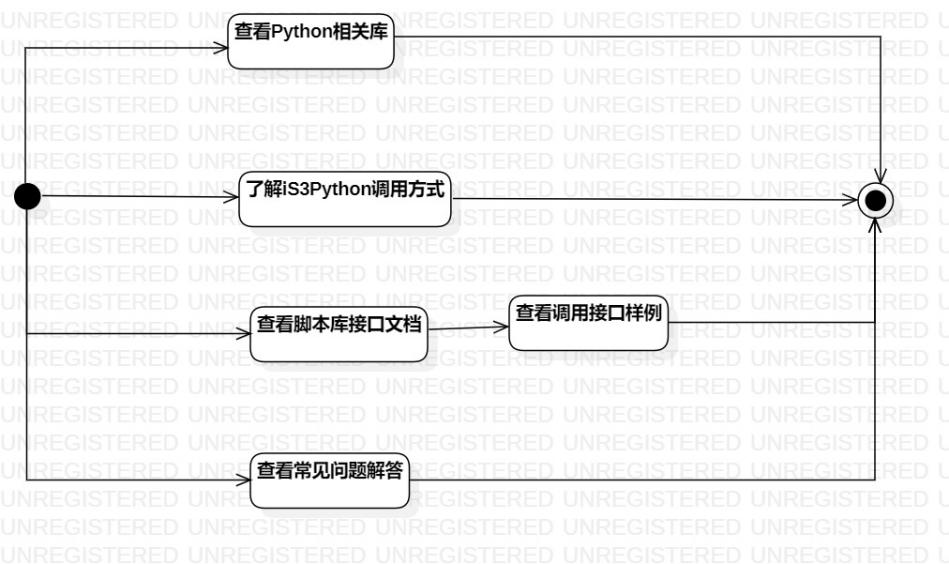
描述：开发iS3平台的工具面板，为用户提供更多的工具使用。

（3）Extend DGObject



描述：在iS3已有的数字化对象的基础上，拓展新的数字化对象类以满足业务场景需求。

（4）Write Functions



描述：编写Python脚本函数，供用户调用。

3. 非功能需求

3.1. 性能要求

* + 1. 用户体验

本项目的最终产出是一份关于iS3平台使用、配置和二次开发的说明文档，该文档应挂于GitHub上在线展示。作为一份说明文档，内容应该简洁、明了、易懂，逻辑严谨、思路清晰，措辞生动准确，尽可能地让用户能够从文档直接获取到想要的平台信息。与此同时，文档的发布平台也需仔细选择，应满足界面优美、交互设计友好等要求，以方便用户查找、搜索想要了解的内容或字段。

* + 1. 时间特性要求
* 响应时间：

在网络状况良好的情况下，对于页面跳转的操作，响应时间应在 1～2s 之内。在线文档的响应时间只要取决于用户的网络状态以及网络对访问GitHub网站的支持。

* 更新处理时间：

iS3数据库的更新涉及网络数据传输，取决于网络传输速度。但正常情况下，数据库更新时间不应超过 1s，网络状况较差的情况下，不应超过 4s。

* 数据转换时间：

首次打开加载项目时需要连接服务器，时间较长，但不超过 3s。在iS3平台中，向服务端发送的请求结果会缓存到本地，第二次加载时能够快速打开。

3.2 安全及保密性要求

* + 1. 工程项目数据

iS3平台展示的工程项目数据来源于开发人员配置的服务端，信息来源稳定可靠，保密性高。系统将对这些数据进行加密存储，严格防止工程数据信息泄露。

* + 1. 在线文档

项目最终提交的文档遵循GitHub开源协议，托管在GitHub上，有关人员可以对其进行补充和修改，以推动iS3平台的开发。

3.4. 灵活性要求

1. 项目的开发采用增量+迭代模型，迭代、分批次完成项目的分析和设计。我们团队以3天为一个 sprint，定期开Daily 会议，并进行 Retrospective 总结会议。采用 Worktile辅助管理开发进程，依照每个sprint 的实际 Velocity 制定下一个 sprint 的工作计划。
2. 采用 Git 版本控制系统进行版本控制。

3.5. 其他专门要求

* + 1. 可维护性

可维护性是指在不影响系统其他部分的情况下修改现有系统功能中问题或缺陷的能力。本项目涉及的代码都有详细注释，并且生成了对应的详尽技术文档。 对于系统开发过程可能出现的报错，我们也在项目文档中也进行了详细说明。

* + 1. 灾难恢复

本项目最终开发的项目代码和文档托管在GitHub云平台上，由GitHub第三方负责数据备份并保证数据的安全性。即使发生灾难造成数据丢失，系统立即通过备份数据进行恢复，并且重新备份，保证系统中始终保持至少一份数据备份。

4.运行环境规定

4.1 设备

iS3 主程序使用C#和Python语言，并基于.Net 4.5 Framework 框架开发而成。建议运行环境如下：

1. 64 位 windows 操作系统
2. 系统至少为 Windows 7 SP1
3. 处理器至少 1.6 GHz ，4 GB RAM， 并具备 11 GB 可用硬盘空间。
4. 详情请见 SRS 2.2 节运⾏行环境部分。

4.2 普通接口

1. 账户授权协议:OAuth 2.0
2. ⽹络访问协议:TCP/IP,HTTP
3. GitHub账号第三方