学生姓名：王禾雨

讲师姓名：刘劼

课程编号：CS33852

日期：2020/7/15

以进程为基础的离散事件仿真框架：SimPy研究报告

SimPy 是一个基于Python语言的以进程(Process)为基础的离散事件仿真框架，在众多工业场景，例如物流、工厂、餐饮等，有较为广泛的应用价值。作为Python语言的第三方开源框架，SimPy具有易于理解、方便实用、成本低廉等优势，是工业场景下简单离散事件模型的强力工具。

1. 发展历史

根据官方文档介绍，开发SimPy的想法最早基于用Python再现并整合Simula以及Simscript两个仿真模拟包，最终呈现的SiPy包以Simula为原本，SimPy包则以Simscript为原本。

SimPy于2002年正式在SourceFroge上发布最初的0.1.2版本。该版本以先进先出(FIFO)、共享资源、和数据分析监控为主要特性。2013年，SimPy 3发布。在这次重大升级中，代码被完全重写，API与代码结构被极大简化 ，优化了基于Python的协程(coroutine)进行异步编程(asynchronous programming)的仿真框架，使得SimPy更加灵活、易用、高效。

1. 理论基础

模拟离散事件系统有三种基本的编程范式，第一种为面向活动(activity-oriented)范式。在该范式下，程序每经过一个固定周期检视各事件发生的状态。由于需要尽可能缩小固定周期以更好的模拟效果，该范式的程序运行会耗费大量的时间。第二种为面向事件(event-oriented)范式，该范式的核心是事件集(event set)，用以保存所有待解决事件(pending event)，程序将选取最近的既定事件来更新模拟系统的时间。这种方法在离散事件数量较多的情况下较为低效，并且只能模拟离散的状态变化，而无法很好的表示连续的过程。

SimPy采取第三种更为高效、灵活的范式——面向进程(process-oriented)。每一项拟真活动将以进程的形式运行，在现代计算机中也多以线程(thread)的形式实现，模拟系统的时间则根据事件更新跳跃式更新推进。SimPy通过应用Python的生成器(generator)，将多个进程以协程的形式实现。协程是Python语言下多进程编程更为灵活高效的实现，可以实现进程的“暂停”和“重启”，使得不同离散事件间的切换、交流、与资源共享更为简洁清晰。

1. 核心概念

SimPy最为核心的三个组件是Environment,

对于超过 4 行的引用，请将引文从左边缘缩进一英寸，勿使用引号。若要应用此格式，请在“开始”选项卡的样式库中，单击“引用”。对于较短的引用，可将其用引号括起，并直接合并到文本中。

表 1

此表标题使用名为“表标题”的样式，“开始”选项卡上的样式库中提供了该样式

| 列标题 | 更多 列标题 | 更多 列标题 | 更多 列标题 |
| --- | --- | --- | --- |
| 行标题 | 表格文本 | 表格文本 | 表格文本 |
| 行标题 | 表格文本 | 表格文本 | 表格文本 |
| 行标题 | 表格文本 | 表格文本 | 表格文本 |

来源： 此来源文本使用名为“表格来源”的样式，“开始”选项卡上的样式库中提供了该样式。

1. 此注释文本使用名为“表格注释”的样式，“开始”选项卡上的样式库中提供了该样式。表格注释使用小写字母（而非阿拉伯数字），以便区别于对正文内容的注释。



图 1。此图表标题使用无缩进样式，“开始”选项卡上的样式库中提供了该样式。使用简略的“图”及图片编号标记图像。

引用作品从其自身页面开始列出后续内容。像键入论文的任何文本一样键入文本内引文即可。请参阅本段末尾所示的示例引文。另请注意，MLA 引文和引用规则广泛适用。因此最好参阅 MLA 第 7 版以了解详细信息。（作者姓氏页面）

若要在所有布局和格式（如悬挂缩进）下查看本文档，请在“视图”选项卡上单击“阅读视图”。若要在创建论文大纲时使用此模板，请在“开始”选项卡上的样式库中单击“无缩进”。

有关设置论文格式的更多指导，请参阅 MLA 第 7 版并咨询导师。

引用作品

作者姓氏，名字。引用书籍的标题。引用名称： 出版商名称，年份。媒体类型（如出版物）。

姓氏，名字，中间名。“文章标题。” 期刊标题（年份）： 页面范围。打印。