

# 需求文档结构化处理领域分析报告

**团 队 名 称 TEAM**

**团 队 成 员 BY1817164 孙竖敬**

**团 队 成 员 SY1806116 赵健宏**

**团 队 成 员 SY1806611 张鹏程**

**团 队 成 员 ZY1806106 刘 良**

**2018年10月**

**一、项目要求阐述**

项目原始要求：给定自然语言文档，设计一套标签，手工嵌入到需求文档中，使其成为流程式的需求。并使用机器学习方法来自动插入标签。然后根据标签来提取信息，形成结构化的需求。

在此要求的基础上，明确给定的自然语言文档按照GWT形式进行描述，产生的结构化需求描述遵循RUCM格式并组成完整文档。

本项目立足软件需求描述方法，旨在建立一个将GWT形式描述的需求文档进行自动化处理转换为RUCM文档的实用系统。RUCM的需求描述方法相比GWT描述形式既保留了自然语言的表达能力与易懂性，又减少了表达的随意性与二义性。项目预期主要应用在软件工程的需求描述环节，在实现上要综合GWT需求文档与RUCM文档的特点设计文档标签，使用机器学习方法实现标签自动嵌入，由机器根据标签提取信息，转换成RUCM形式的需求文档。

项目要求的具体条目有

1. 给定自然语言需求文档：自然语言描述的需求文档是系统的输入数据。给定的自然语言文档以GWT形式描述。在软件开发过程中，需求文档记录了需求分析和导出的结果，是对用户和系统需求的全面详细的描述，自然语言的描述方法是最直观易懂且应用广泛的需求描述方法，但是自然语言的二义性使得这种方式有着明显的弊端，GWT格式的需求描述相比一般的自然语言描述方式准确性更高，但仍然可以转换成结构化程度更高的描述方式。
2. 设计一套标签：标签可以较高的层次上对需求文档的结构进行抽象，反映出软件需求的流程特性，在本项目中，需求文档标签是GWT描述与RUCM描述之间的桥梁，通过标签标注出GWT描述中与RUCM描述对应的实体与关系，在此基础上可以迅速地根据标签抽取信息产生RUCM文档。。
3. 使用机器学习方法自动插入标签
4. 使用机器学习方法：使机器能够对自然语言描述的需求文档进行处理需要自然语言处0理技术的支撑，在本项目中主要体现为自然语言的信息抽取（Information Extraction）过程，作为信息抽取的子过程，分词、标注、分块等文本处理操作都可以结合机器学习技术，通过引入机器学习方法可以得到准确性更高、灵活性更好的文档处理工具。
5. 自动插入标签：自动插入标签是项目中应用机器学习方法的目的，通过对自然语言描述的处理，在对描述内容达到一定理解程度的基础上可以由机器自行判断文档标签的位置，特别是在在处理规模较大的文档时可以拥有高效率。
6. 根据标签提取信息：标签完成了对原始的自然语言描述的结构划分，显示出了需求的流程特点。在为文档插入标签后，GWT描述中的RUCM内容被标记出来，根据标签可以快速提取出文档中对应于RUCM描述的内容。
7. 形成结构化的需求：在通过标签提取出信息后，提取出的信息遵循RUCM描述格式，在此基础上，可以将信息保存为RUCM描述的需求文档，获得结构化的需求。

**二、领域定位**

项目所属的具体领域在现实世界中有真实系统存在的，具体来说，项目所属的领域是软件工程中需求工程领域的需求描述环节，并且，根据项目的具体要求，需求文档的处理过程中，还会涉及到其他相关领域，例如，标签、AI、自然语言处理以及对软件提出不同具体要求的不同行业领域等。

对相关领域定位的具体分析如下。

软件工程-需求工程-需求描述环节：由于软件开发具有复杂性、一致性、可变性、不可见性的特征，软件开发往往面对着客户不满意、风险和成本问题、项目过程失控、无力管理团队等挑战。软件工程一直致力于探索软件开发问题的解决之道，将系统性的、规范化的、可定量的方法应用于软件的开发、运行和维护，即工程化应用到软件上。著名软件工程专家B.Boehm综合有关专家和学者的意见并总结了多年来开发软件的经验，于1983年在一篇论文中提出了软件工程的七条基本原理：（1）用分阶段的生存周期计划进行严格的管理。（2）坚持进行阶段评审。（3）实行严格的产品控制。（4）采用现代程序设计技术。（5）软件工程结果应能清楚地审查。（6）开发小组的人员应该少而精。（7）承认不断改进软件工程实践的必要性。其中，需求工程在软件工程中的重要性是毋庸置疑的。需求是人们要解决的某个问题或达到某种目的的需要，是系统或其组成部分为满足某种书面规定（合同、标准、规范等）所要具备的能力。需求将作为系统开发、 测试、验收、提交的正式文档依据。需求不只是软件过程的阶段之一，同时也涉及到软件工程的方方面面。本次项目的领域可以具体到需求工程的需求描述环节，即将原始的GWT需求描述转变为RUCM标准格式的需求描述。这种标准格式的描述可以便于人们对于需求的统一的理解，也可以用以后机器对于需求文档地进一步转化过程，如转化为模型或转化为设计等。

在需求描述环节，自然语言描述是最初被用来描述软件需求的方法，并且也是需求描述使用最广泛的方法，它易于表达，直观且通用，但是往往具有二义性，具有不同专业背景的读者对同一个需求描述可能会得到不同的理解。结构化描述要求按照一种标准方式来用自然语言表达需求，这种方式保留了自然语言描述的好处，包括表达能力和易懂性，同时对描述施加了一致性的约束。使用结构化的RUCM描述方法来描述系统需求，以表格形式突出了需求的主要内容，清晰地描述了各个需求间的关系。

标签：标签是无等级的关键字或分配某项物品的信息（如互连网书签，数字图像，或电脑档案）。这样的数据有助于说明一个项目，并允许它再次发现浏览或搜寻。根据系统，标签本身被非正式创作者或其他观察者选择。 在许多用户标记许多项目的网站上，标记的汇集成为分众分类法。本项目中的标签类似于html标签，即是对某一类文本进行定义的标签，标识出原始的需求描述中与RUCM表项相关的信息，是在分析理解文本的基础上对文本内容的划分。

AI：即人工智能（artificial intelligence），人工智能一般指通过普通计算机程序的手段实现人类智能技术。其核心问题包括建构能够跟人类似甚至超越的推理、知识、规划、学习、交流、感知、移动和操作物体的能力等。针对这些子领域中让机器学会“学习”的问题，派生而出了机器学习的方法，而机器学习已经成为了人工智能领域的一个重要的分支。机器学习理论主要是设计和分析一些让计算机可以自动“学习”的算法，可分为监督性学习、非监督性学习和强化学习。根据项目要求，对标签的自动插入需要使用机器学习的方法。而机器学习的方法选取将直接影响到标签插入的效果，最终将决定对需求文档的处理结果。

自然语言处理：是人工智能和语言学领域的分支学科，研究能实现人与计算机之间用自然语言进行有效通信的各种理论和方法。自然语言理解方向，主要目标是帮助机器更好理解人的语言，包括基础的词法、句法等语义理解，以及需求、篇章、情感层面的高层理解。在本项目中，给出的输入是自然语言写的文档，而对自然语言的处理有着种种困难，尤其是要消除歧义的问题，让程序正确理解需求文档。这就需要参考自然语言处理中对自然语言理解所采用的方法，来保证自然语言文档描述的需求能够被程序正确识别。

对软件提出不同具体要求的不同行业领域：需求会根据干系人的不同而不同，而干系人往往会由于所在领域的不同而不同。所以在具体的软件开发项目中，软件需求与约束会表现出较强的行业特性会与很多方面相关，如政府[机构](https://baike.baidu.com/item/%E8%A1%8C%E4%B8%9A%E5%88%86%E7%B1%BB/2847868?fr=aladdin#2_1)、医药卫生、交通运输、信息产业、生活服务、教育教学、游戏社交等，应用于不同行业的软件产品需要充分考虑与行业相适应的需求。

**三、主要术语及解析**

1 软件需求相关术语

1. 需求工程（Requirements engineering，RE）：指在工程设计过程中定义、记录和维护需求的过程。
2. （需求的）符合性（Conformity（of requirements））：需求规格说明书符合某些标准的程度。
3. （需求的）一致性（Consistency（of requirements））：对一套需求的描述无前后矛盾程度。
4. 约束，边界条件（Constraint）：一种需求，它限制解决方案在那些必要的满足给定的功能需要和质量需求之外的空间。
5. 实体（Entity）：一般而言，实体是指一个元素或元素的集合，它可以表示任何可想象的产品，例如，一个系统、现实的一部分、一件东西、一个组织、-一个过程等。在实体关系建模中指单个对象，它具有一个标识，并且不依赖于另一个对象。
6. 模型（Model）：一个存在的现实或一个将 要创建的现实的抽象表达。广义的模型是现有实体或将要创建的实体的表达，而实体表示现实的任何部分或任何其它可能的元素或现象，包括其它模型的组合。就模型而言，实体称为原件。在需求工程中，需求可用模型来说明。
7. 功能需求：包括对系统应该提供的服务、如何对特殊输入做出反应，以及系统在特定条件下的行为的描述。在某些情况下，功能需求可能还需明确声明系统不应该做什么。
8. 非功能需求：指不直接关系到系统向用户提供的具体服务的一类需求，或对系统提供的服务或功能给出的约束。包括时间约束、开发过程的约束和所受到的标准的约束。非功能需求经常适用于整个系统而不是个别的系统功能或服务。
9. 产品需求：这些需求定义或约束软件的行为。包括系统执行速度和内存消耗等性能需求，包括指定系统可以接受的出错率等系统可靠性需求，也包括信息安全性需求和可用性需求。
10. 机构需求：指广泛的系统需求，起源于客户所在的机构和开发者所在的机构中的政策和规定。包括过程标准，即机构中采用的过程标准；实现要求，如所采用的程序设计语言和设计方法；交付需求，即有关对产品及其文档交付的要求。
11. 外部需求：包括所有来自系统外部因素和开发过程的需求。如监管需求；立法需求和道德需求等。
12. 领域需求：系统的领域需求是从系统的应用城中得出的而不是从个别系统用户那里得到的。它们可能本身就是新的功能需求，也可能是约束现有的功能需求，或者是特别的计算必须如何进行的规定。
13. 软件需求文档（软件需求描述或SRS）：是对系统开发者需要实现的功能的正式陈述。它应该包括系统的用户需求和一个详细的系统需求描述。
14. GWT格式：即Given，When，Then。描述内容如图1所示。

图1 GWT描述规范



图2 RUCM文档描述规范

1. RUCM格式：即受限用例建模（Restricted use case modeling），

如图2所示，包含必选的用例名、简述、前置条件、主次活动者、 依赖、泛化、基本事件流信息和可选的特定分支流、全局分支流 及边界分支流。

2 机器学习相关术语解析

1. 机器学习：机器学习是用数据或以往的经验，以此优化计算机程序的性能标准。机器学习目前已广泛应用于数据挖掘、计算机视觉、自然语言处理、生物特征识别、搜索引擎、证券市场分析、战略游戏和机器人等领域。
2. 深度学习：深度学习（deep learning）是机器学习的分支，是一种试图使用包含复杂结构或由多重非线性变换构成的多个处理层对数据进行高层抽象的算法。深度学习是机器学习中一种基于对数据进行表征学习的算法。深度学习的好处是用非监督式或半监督式的特征学习和分层特征提取高效算法来替代手工获取特征。
3. 监督学习：从给定的训练数据集中学习出一个函数，当新的数据到来时，可以根据这个函数预测结果。监督学习的训练集要包括输入和输出，也可以说是特征和目标。训练集中的目标是由人标注的。常见的监督学习算法包括回归分析和统计分类等。
4. 无监督学习：无监督学习与监督学习相比，训练集没有人为标注的结果。常见的无监督学习算法有生成对抗网络（GAN）、聚类等。
5. 半监督学习：介于监督学习与无监督学习之间的学习方式。
6. 人工神经网络（Artificial Neural Network，ANN）：简称神经网络（Neural Network，NN）或类神经网络，在机器学习和认知科学领域，是一种模仿生物神经网络（动物的中枢神经系统，特别是大脑）的结构和功能的数学模型或计算模型，用于对函数进行估计或近似。神经网络由大量的人工神经元联结进行计算。大多数情况下人工神经网络能在外界信息的基础上改变内部结构，是一种自适应系统，通俗的讲就是具备学习功能。现代神经网络是一种非线性统计性数据建模工具。

3 自然语言处理相关术语解析

1. 自然语言处理：（Natural Language Processing，NLP）是人工智能和语言学领域的分支学科。此领域探讨如何处理及运用自然语言；自然语言处理包括多方面和步骤，基本有认知、理解、生成等部分。自然语言认知和理解让电脑把输入的语言变成有意义的符号和关系，然后根据目的再处理。
2. 中文分词：中文分词(Chinese Word Segmentation) 指的是将一个汉字序列切分成一个个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。在英文的行文中，单词之间是以空格作为自然分界符的，而中文只是字、句和段能通过明显的分界符来简单划界，唯独词没有一个形式上的分界符，虽然英文也同样存在短语的划分问题，不过在词这一层上，中文比之英文要复杂得多、困难得多。
3. 词性标注（Label）：机器理解自然语言文本时往往需要对词的词性进行标注。标注用于表征词的一种隐状态，隐藏状态构成的转移就构成了状态转移序列。例如：苏宁易购/n 投资/v 了/u 国际米兰/n。其中，n代表名词，v代表动词，n、v都是标注，以此类推。
4. 信息抽取或信息提取（Information Extraction）：通常处理对象为非结构化文本，在一定程度分析和理解文本的基础上，从大量文字数据中自动抽取名称、事件等特定消息（Particular Information）。
5. 自动摘要（Automatic Summarization）：从一个或多个文本中自动摘取包含了原文中最重要信息的部分，通常用于提供已知领域的文章摘要，例如产生报纸上某篇文章之摘要。
6. 依存句法分析（Dependency Parsing, DP）：依存句法通过分析语言单位内成分之前的依存关系解释其句法结构，主张句子中核心动词是支配其他成分的中心成分。而它本身却不受其他任何成分的支配，所有受支配成分都以某种关系从属支配者。

**四、领域系统架构分析**

从领域系统架构分析角度，类图和状态图相结合可以清晰地解释系统架构模型，所以我们在此部分通过对构建的图3类图和图4状态图进行分析。

**类图分析：**

根据要求，本项目的工作是对输入（符合Given-When-Then要求的需求文档）进行处理，使之成为符合RUCM规范的需求文档，故而领域系统的主体部分是对需求文档的处理，次要部分则是输入与输出。我们用类图表述领域系统中各个组成元素以及元素间存在的静态关系，用状态图表述主要元素在领域系统中的生存期的动态行为，从而清晰地反映整个领域系统的架构。

类图分析：

按照项目功能需求，整个领域系统被分成输入、处理与输出三个部分，每个部分的所包含的元素、每个部分内部元素之间的关系与三个部分之间的关系如图所示（图3）。

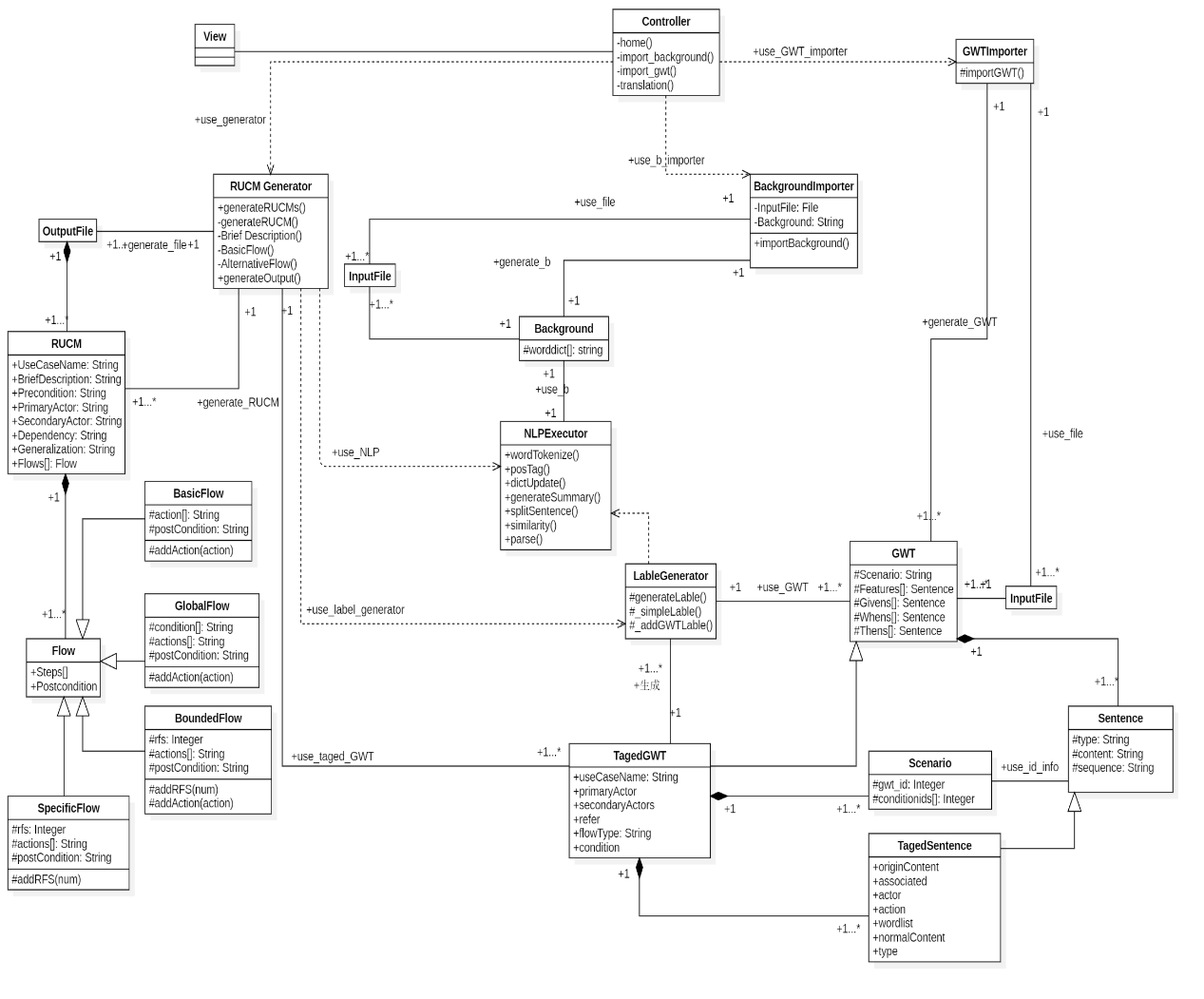


图3 类图

1. 输入部分：

在输入部分中，主要完成对给定文档预处理的工作，使之成为符合系统输入规范的输入单元。本部分涉及到的元素包括：GWT文档、领域背景以及GWT描述。

1. GWT文档是给定的符合Given-When-Then要求的文档，由于给定的初始文档不一定是符合规范的文档，故而需要用预处理器对之进行处理，使之成为严格符合格式的GWT描述。
2. GWT描述是本部分的输出，同时也是下一部分的输入，由Feature、Scenario、Given、When和Then五个元素组成。
3. 由于各个领域具有不同的专业术语，故而需要GWT文档所对应的领域背景信息来提高文档处理的精度。领域背景部分通过对输入需求方导入的领域背景进行处理，生成领域术语关系图，获得领域背景信息
4. 处理生成部分

本部分是本项目中的核心。在本部分中对GWT描述的处理采用的方法将直接影响到最后生成的RCUM需求文档的质量。整个部分以自然语言处理——标签标注——信息提取的流程进行，以机器学习的方法进行，具体涉及到的元素为：自然语言处理器NLPExecutor、中间标签生成器LableGenerater、RUCM生成器RUCM Generater、中间标签对象TaggedGWT、RUCM对象。

1. 自然语言处理器的主要功能是为中间标签生成器和RUCM生成器提供支持，自然语言处理器提供的主要操作有领域背景信息导入、分句、分词、词性标注、句法分析、句子相似度、摘要生成。其中领域背景信息导入为自然语言处理器拓展领域知识，提高自然语言处理的准确度；分句操作将一段文本拆分成有意义的句子；分词操作将一句话拆分成词的集合；词性标注在分词的基础上标注每个词的词性；句法分析根据分词和词性标注结果生成句子的语法树；句子相似度计算采用向量化方法计算两个句子结构的相似度；摘要生成操作根据一段文本生成对文本的摘要描述。
2. 中间标签生成器以自然语言处理器为支持处理输入的原始GWT数据，根据标签的设计逐个分析出标签内容，依次添加GWT级别和句子级别的标签，输出中间标签对象。
3. 中间标签对象是生成RUCM的基础，既包含了原始GWT的全部内容，也针对生成RUCM的需要使用了标签，通过中间标签生成器将原始GWT对象转换成中间标签对象，提供RUCM生成器使用。
4. RUCM生成器用于产生系统的最终输出，根据RUCM规定的格式从中间标签对象中提取信息填写RUCM的对应字段，对于不能直接提取的字段应用自然语言处理器分析中间标签对象进行生成，产生RUCM对象，将RUCM对象输出到文件中。

**状态图解释：**

本次项目所需要处理的主要对象是文档，贯穿着项目的始终。在输入状态完成后，初始的文档被转化成了一系列的GWT描述，而本项目的关键就在于GWT描述的处理上——将其转化为RUCM需求文档。在整个转化过程中，先对输入的GWT文档进行分析，对其中的GWT描述生成对应的GWT对象，通过中间标签生成将GWT对象转换为中间标签对象TaggedGWT，由RUCM生成器将TaggedGWT对象转换成对应的RUCM对象并产生输出文件。

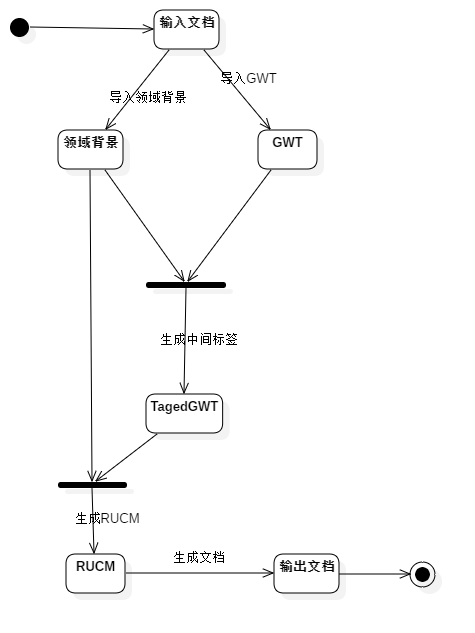


图4 词性标注状态图

1. 输入文档：在此状态下有两种输入文件，一种是记录了一组GWT描述的纯文本文件，以utf-8方式编码；一种是记录了GWT描述领域背景信息的纯文本文件，以utf-8方式编码。
2. 领域背景：在此状态下系统应用领域背景信息文件更新自然语言处理器，提高自然语言处理器的准确度。
3. GWT：在此状态下系统分析输入的GWT文档，生成对应的原始GWT对象用于后续的处理。
4. TaggedGWT：在此状态下系统对一组原始的GWT对象进行自然语言处理和标签生成，产生一组对应的中间标签对象。
5. RUCM：在此状态下系统根据RUCM的格式从一组TaggedGWT对象中提取信息，生成一个对应的RUCM对象。
6. 输出文档：在此状态下系统将一组生成的RUCM对象写入一个纯文本文件中，输出最终的结果文档。

**五、领域系统运行环境**

将该领域系统看成一个黑盒，与该系统发生交互的系统主要可以分成文档、软件支持设施与人员三大类。其中，文档可以粗略地分为为GWT文档、RUCM文档两类，软件支持设施可以粗略地分化为软件运行硬件设施、软件运行网络设施两类，人员分为需求提出方和软件开发方两方，软件开发方通常具有产品经理、开发人员等主要工作人员。在下面，本文将按照以上分类分别进行描述：

1. **文档**
2. GWT文档：GWT文档是本系统唯一种类的输入文档，占据十分重要的环境地位。该类文档本身就具有一定结构，是按照Given-When-Then格式进行描述。其中，Given部分描述了执行行为之前的状态是什么，When部分描述了所指定的行为，Then部分描述了你通过以上行为希望产生什么样的改变。GWT在通常情况下应该是表述清晰、简单易懂的文档，但是在实际情况中可能不是这样。提出需求的用户可能不习惯GWT文档的描述方式，有可能采用一些不符合GWT文档的描述方式。以上这种情况可能会根据提出需求的用户的不同而不同，所以我们先假设这种情况出现的频率比较少，以简化系统的设计难度。
3. RUCM文档：RUCM文档是本系统最重要的结果文档，是本系统的设计目标。该文档也是结构性的文档，其描述更加严谨，将用于架构师的设计工作与程序员的编码工作。这类文档，我们希望它的结构受限，不会发生改变。RUCM文档格式如下：
   * 1. Use Case Name：用例名称，通常以动词开头。
     2. Brief Description：用例内容的简要描述。
     3. Precondition：用例的前置条件，即用例执行前必须满足的条件。
     4. Primary Actor：主要活动者，即出发用例执行的活动者。
     5. Secondary Actors：次要活动者，除首要活动者外系统所依赖的其他活动者。
     6. Dependency：依赖关系，描述与其他用例之间的包含和拓展关系。
     7. Generation：泛化关系，描述与其他用例之间的泛化关系。
     8. Basic Flow：基本事件流，描述完全顺利执行用例时的事件流程。
4. Steps（numbered）：事件流步骤
5. Post Condition：基本流的后置条件
   * 1. Specific Alternative Flows：特定分支流
6. RFS：引用基本流程步骤的名字与步骤号，例如RFS 3
7. Steps（numbered）：事件流步骤
8. Post Condition：分支流的后置条件
   * 1. Global Alternative Flows：全局分支流
9. Guard Condition：进入该全局分支测试流程执行的守护条件
10. Steps（numbered）：事件流步骤
11. Post Condition：分支流的后置条件
    * 1. Bounded Alternative Flows（Test Sequence）：
12. RFS：引用步骤的所在事件流的名字与步骤号
13. Steps（numbered）：事件流步骤
14. Post Condition：分支流的后置条件

由以上RUCM文档格式可知，其中有些信息可以从单个GWT文档中提取或分析得出，有些则需要多个GWT文档结合进行分析。

1. **软件支持设施**
2. 软件运行硬件设施：这部分涉及到软件运行所需要的芯片、显示器、操作装置等。这是操作该软件系统的主要环境，其鲁棒性与环境变化的稳定性将决定系统操作的舒适度与稳定性。是该系统重要的基础设施。但是这不是我们所能决定的因素，而是由用户决定在构建基础设施的投入力度所决定。所以我们应力所能及地减少对于软件支持设施的需求。
3. 软件运行网络设施：这里涉及到用户网络是开放网络还是私有网络，是高速网络还是慢速网络的问题。同时网络环境会经常发生变化，我们需要考虑到网络环境的变化会对我们的系统产生什么样的影响。同时，需求文档有时是属于一个公司的机密内容，如果被友商获取可能会导致种种严重问题，例如市场占有度降低，用户学习成本增大，友商抄袭等。所以，该系统对于联网问题应严肃对待，同时注意数据保护环节。
4. **人员**
5. 需求提出方：此为通常情况下提供GWT文档的人员，是需求的产生方，是本系统输入的来源方。其具有领域多样化，状态变化频度高等特征。为了使系统能够更准确地辨识GWT文档中专业性较强的词汇与描述，系统将需要导入专业词汇库等必要的领域背景信息，需求提出方同时也是领域背景信息的提供者。需求提出方可能来自于这个世界的各个行业，而提出的需求又很可能是在这个世界上尚未存在的事物。这给我们对于GWT需求文档的分析带来了很多的挑战。同时，通过需求提出方对于GWT文档的了解程度不同可分为三类：
6. 不知道什么是GWT文档：这种情况下，系统对于需求文档的分析难度将会非常巨大，可能需要产品经理手动将原始需求文档转化为GWT表述格式。
7. 了解过什么是GWT文档：这种情况下，需求文档有时可能是以GWT文档格式进行书写，但是可能会出现文不对题的情况。在此情景下，系统对于该类文档尚存在一些分析能力，但是会有转化不准确，易读性变差等问题。
8. 掌握GWT文档书写方法：在这种情况下，本系统力求得出最佳的RUCM结果，及RUCM文档具有准确性高，易读性高等特点。
9. 软件开发方：在系统过程中，软件开发方主要的参与者通常有产品经理和开发人员，在需求提出方提供GWT文档后，软件开发方工作人员是转化出的RUCM文档的主要使用者。
10. 产品经理：此为RUCM的需求者之一，并且可以对于RUCM的格式进行一定的定制化操作。其对于输出的要求决定了本系统的设计目标。产品经理是每个产品牵头人，在市场营销部，对某个产品在集团内的盈亏负责，并为这个产品的运作去协调所有的人，并充分地协调这个产品的所有运作环节和经营活动。一般来说，产品经理是负责并保证高质量的产品按时完成和发布的专职管理人员。他的任务包括倾听用户需求；负责产品功能的定义、规划和设计；做各种复杂决策，保证团队顺利开展工作及跟踪程序错误等，总之，产品经理全权负责产品的最终完成。另外，产品经理还要认真搜集用户的新需求、竞争产品的资料，并进行需求分析、竞品分析以及研究产品的发展趋势等。综合以上产品经理的特征，我们可知产品经理需要在尽量短的时间内对于一个系统有足够充分和完整地了解，所以本系统的生成速度和准确性需要有足够的保障，同时还需要保障RUCM文档的可读性。
11. 开发人员：此为RUCM文档的另一位需求者，开发人员需要按照RUCM对系统需求的描述对项目总体和接口进行把握，在此基础上制定技术实现方案，完成对项目的代码实现工作。在这里，系统转化输出的RUCM文档的准确性和易读性对开发人员的工作会产生很大的影响，RUCM描述的质量需要得到保证以方便开发人员的工作。

以下两幅图是以系统作为黑盒，对于与其接触并产生数据交换的人员的解析图片，图5是用例图，图6是信息交互图。

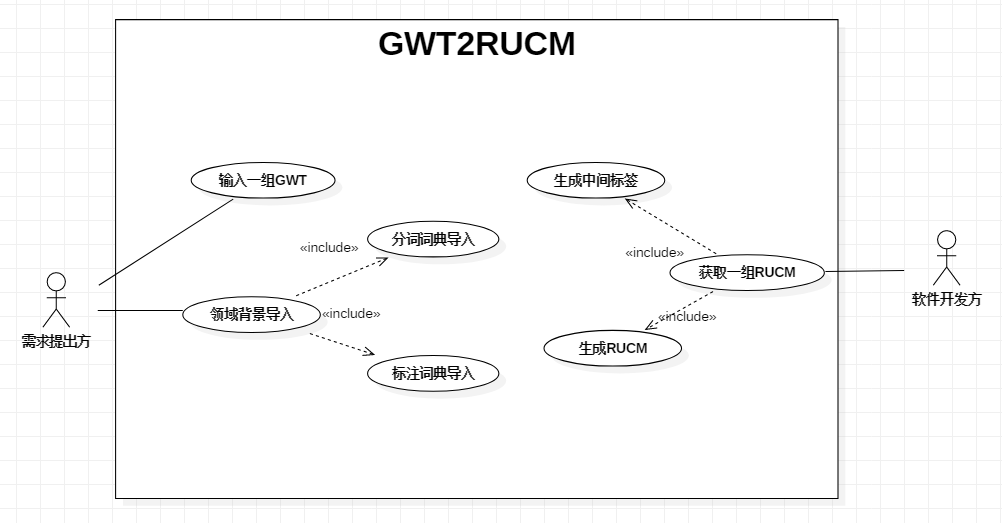


图5 用例图

GWT2RUCM

需求提出方

产品经理

开发人员

GWT文档

RUCM文档

RUCM文档

领域背景信息

图6 信息交互图

**六、领域系统主要流程分析**

**领域背景信息以及GWT文档导入流程**

本系统旨在将输入的GWT格式的需求文档转化成RUCM格式的结构化的需求文档。针对GWT文档涉及领域不同，在后面的信息处理阶段生成的结果将会有很大不同。所以针对GWT文档的特定背景，我们需要向系统导入该领域的背景信息。所以本系统应在领域背景信息确定的情况下才能够对所需转换的GWT文档进行处理。

1. 领域背景信息导入：本流程目的在于将需求提出方提供的领域背景信息导入到系统生成产品领域背景数据库。在本流程中，系统会判断导入的领域背景信息是否充足，只有在领域信息背景信息整理充足并导入到系统的情况下，系统才会进行下一步：通过领域信息生成领域术语关系图。最终系统将输入信息设置成产品领域背景数据库。
2. GWT文档导入：在领域信息背景信息导入完成的后，系统会提示向系统输入GWT文档，此时本流程即需要执行，用户将与领域背景相对应的GWT文档导入到信息中。

**信息预处理流程**

对于输入的GWT格式需求文档需要进行预处理流程，本流程中主要的目的在于以下三点：剔除无效信息、切分GWT描述、生成规范化的GWT需求文档。

1. 剔除无效信息：本过程目的在于将空白行、空格、重复赘余词语或语句进行剔除。本流程对所给的GWT格式需求文档进行处理，但只能对相对简单的语句格式问题进行修整，并不会对文档内容进行分析，所以是本系统最为简单的流程。作为预处理的最初流程，筛选剔除工作如果完成的太过于粗劣，将会大幅度提升系统处理的工作量，而过于严苛以至于将部分关键信息删除又会在最后的结构化需求文档中丢失用户的部分需求。本流程虽然简单却有着很关键的作用。
2. 切分GWT描述：本过程中针对前面剔除无效信息后的文档按照GWT规范格式进行大概内容的划分，将GWT文档进行切分，将无赘余信息的文档按照GWT标准规范切分成一个个的GWT描述。
3. 生成规范化的GWT需求文档：经过上述两个流程，本过程直接将规范化的GWT格式的需求文档输出给下一流程。

**信息处理流程**

本过程属于本系统的主要流程部分主要对GWT格式需求文档进分词和词性标注两种操作。

1. 分词：首先根据预处理流程对GWT描述的语句复杂性判断，本流程主要对需求文档中复杂语句进行分词操作，主要运用机器学习的方法对已有的语料库进行学习，本流程中需要阅读文献对相关学习模型进行分析和模拟，找到对本流程最合适的学习模型进行相关改造对本系统所给的GWT需求文档的信息进行分词处理，设置一定的学习率，将文档输入到学习模型中进行分词处理，本流程中将会根据GWT文档中的基本信息和GWT描述产生五种词链信息：Feature词链、Scenario词链、Given词链、When词链、Then词链 。
2. 词性标注：根据提取的各种词链，系统会利用训练模型和之前建立的领域背景数据库对各种词链的信息进行词性标注，从而将各词链相对来说比较具有整体性的语句进一步细化，比如在某一词链中利用学习的方法将本语段中的主语、谓语、宾语找到并进行标注，从而提取到本句话的重点，为下一步的关系交互做准备。

在本流程中训练模型的选取需要经过不断调研文献中各种训练模型的特点并且从中提取我们所需要的部分进行改善，所以得到适合本流程操作的训练模型也是实现这一功能的重要部分，在本部分我们就不再进一步详细说明了，将会在报告的下文中详细分析训练模型利用学习的方法进行分词和词性标注的过程。同时本流程中需要将之前系统建立的领域背景数据库对应用到GWT描述中，对其进行分词和词性标注，所以分词和词性标注进行阶段都需访问领域背景数据库，按照该GWT所属领域的分词、标注规范进行相应的操作。

·**信息提取流程**

在本流程中系统将会对经过上一流程处理的GWT文档进行RUCM信息的提取。该信息提取流程主要分为三个并行部分：基本信息提取、关系信息提取、事件流信息提取。

1. 基本信息提取：按照RUCM模板中列出的基本信息对词性标注结果进行信息抽取，依赖于基本信息抽取结果的格式，生成的基本信息抽取结果各个部分为：用例名称、简要描述、用例前置条件、主要活动者和次要活动者。
2. 关系信息提取：根据RUCM对词性标注结果进行信息抽取，本身依赖于关系抽取结果的格式，生成的关系抽取结果由依赖关系和泛化关系组成。
3. 事件流信息提取：根据RUCM模板中列出的事件流对词性标注结果进行信息抽取，依赖于事件抽取结果的格式。事件流抽取结果的基本组成部分为：基本描述、步骤和后置条件，由四种类型事件流组成的：基本事件流、特定分支流、全局分支流和有界分支流。

在本流程中关系所有的GWT描述的基本信息提取结果是通用的，所以我们的顺序图中强调了按照GWT描述语段顺序对事件流信息进行依次提取。通过逐个分析GWT描述，我们可以得到GWT描述之间的关系信息。所以关系信息和事件流信息的提取在同一个循环框中。

**结果生成流程**

上面各个流程的工作已经将需求文档的主要信息提取，该部分可以说是将GWT格式转换成RUCM格式需求文档的流程，结果生成流程主要是利用RCUM描述格式的需求文档各部分对前面提取的信息进行收取划分，GWT格式中的信息在前面流程中已经碎片化，该部分的作用就是将这些碎片系统的归类。系统会根据RUCM各部分所需信息对信息碎片进行收集然后整合成RUCM词链，从而生成RUCM格式的需求文档。

由于GWT文档转换为RUCM文档系统首次提出，并不存在各种资料信息的借鉴，所以我们计划通过和老师、助教讨论建立起一个转换体系，用于将GWT描述信息提取转换为RUCM文档中的相关描述。

下图分别是系统的顺序图和活动图

图5 顺序图

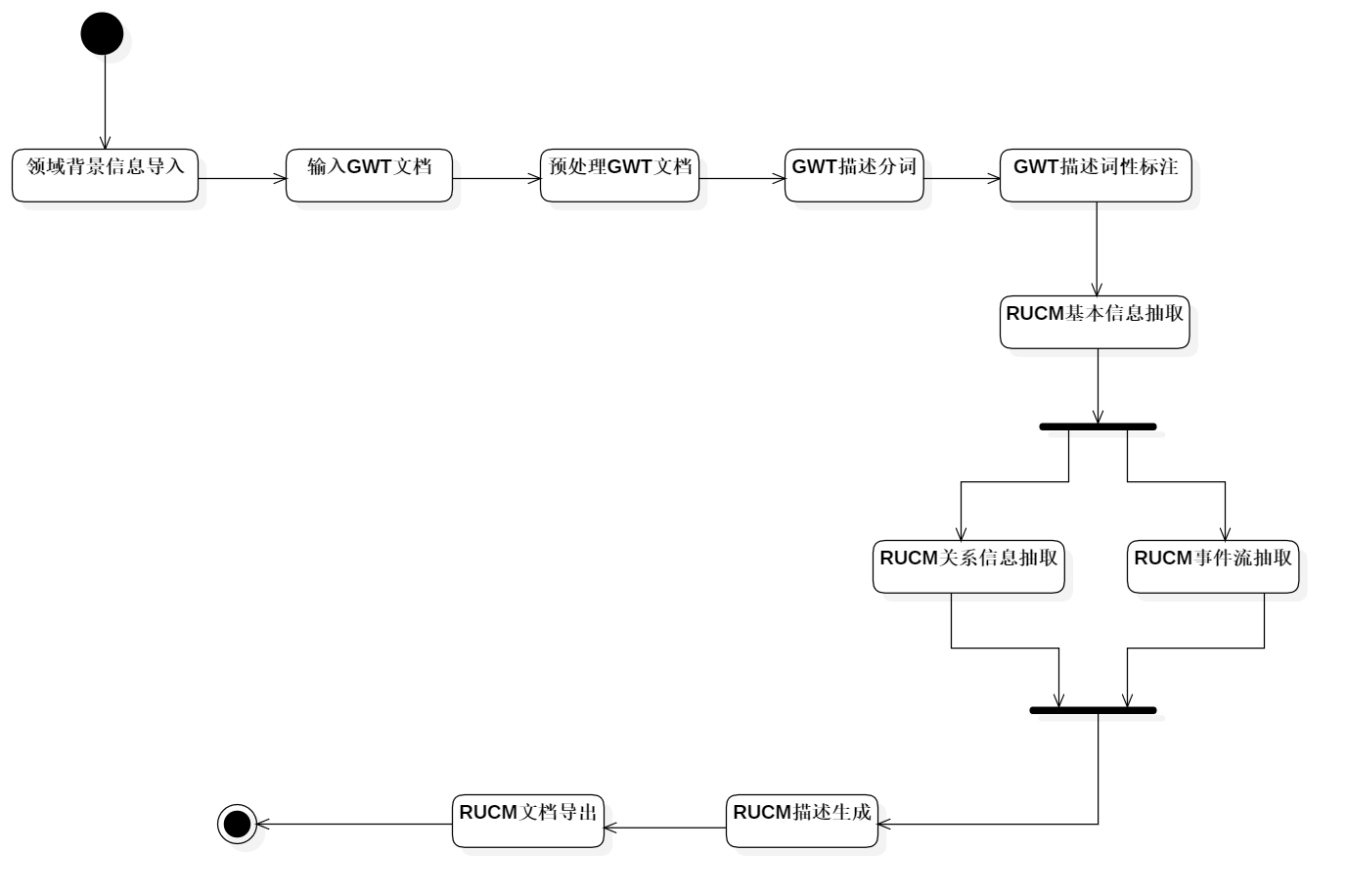


图6 活动图

**七、系统用户识别**

1. 基于对领域系统的环境分析和流程分析，识别领域系统的用户及特征，注意用户可以是自然人，也可以是一个人造系统。
2. 需求提出方：通过以上分析，已知其提供GWT文档作为系统的输入。其对于GWT文档的了解程度将决定系统分析输入的难度。下面我们将对需求提出方展开进行分析。除了依据对GWT文档的了解程度进行分类以外，我们还可以按照行业对需求提出方进行分类等。这些类型的需求提出方，在需求特点、描述准确度、项目完成时间紧迫程度、项目完成达标标准等都各有特点。例如，工程行业对于项目的鲁棒性要求就会比游戏类的要求要高。于此同时，我们可以从安全可靠性要求水平对需求提出方进行分类。对于安全可靠性要求要求高的项目可以牺牲易读性来保证准确性，而对于安全可靠性没那么高的项目可以牺牲准确性来保证易读性，以便快速对项目进行迭代。不仅如此，有些行业可能对于保密性有很高的要求，这就要求我们对于GWT文档进行某些加密或隔离处理。看待需求提出方的角度还有很多，本次领域分析只提出一部分，接下来在需求分析和产品设计的过程中，我们将依据真实环境进行更加详细和多角度的细分和细化。
3. 产品经理：产品经理的简单特征在上文已经有一定的解释，在这里我们将进行更为细化的描述。产品经理作为每个产品的牵头人，对于整个产品和产品过程负责。基于以上原因，我们将给予不同于开发人员的操作权限。其不仅可以查看RUCM文档，还可以设计标签，并设置RUCM文档特征，以及开发人员对于RUCM文档的访问权限。产品经理可以按照工作经验和通常从事领域进行区别。工作经验较少的产品经理可能对于产品整个流程缺少认识，所以本系统将存在默认的设置。同时，产品经理可能之前从事的领域和现在需要做的领域存在差别，在这种情况下，该系统将起到辅助产品经理对于项目整体的认识效果。
4. 开发人员：其将获取RUCM文档，但是不同于产品经理，其对于RUCM不具有修改权限。同时开发人员在访问时也可能受到一些限制，保证项目一定程度上的秘密性。但是通常情况下，开发人员将依据RUCM文档对于系统有一个整体且具体的了解。

6. 机器学习框架：此为系统功能的一部分，但是同时又是系统外部的API和标准。当前存在很多机器学习框架，用来方便开发人员将机器学习落于实地，可以通过预训练来保障开发人员仅仅需要少量的数据就可以得出一个较为可靠的模型。由于现在文档数量的不确定性，我们希望采用这类框架，使用不多的数据就可以产生足够好的结果。

**八、待开发系统的目标分析**

本项目的要求是需求文档的结构化处理，具体过程是将给定的GWT格式的文档用机器学习的方法使之转化为RUCM需求文档。这一项目主要针对软件开发流程中的重要环节——需求分析，故涉及到了客户、开发者对此软件的期望。

针对客户而言，客户提供的信息的是需求描述，期望是软件开发方能充分了解自己的需求，最终开发出来的软件满足自己的需求。作为软件的需求方，客户本身往往来自其他领域，缺乏软件开发领域的专业知识，故而初始需求必定是非形式化的需求，且需求势必充满本领域特色，给软件开发方准确理解客户的需求增添了重重困难。故而通过与客户确定基本的输入格式来施加一些简单约束，一方面不会增加客户学习严格、完整的需求文档的学习成本，另一方面也使需求以相对规整的格式为软件开发人员所理解。Given-When-Then标准作为一种有效且简单的输入规范，客户需要以此格式来提供自己的需求信息。

对于软件的开发者而言，在获得客户输入的GWT格式需求文档后，仍应对此文档进行处理，使之成为严格遵守规范的需求文档。因为客户给出的GWT格式的文档中的需求信息是以集合的形式存在于文档中的各项描述中，所以软件开发人员无法直接将GWT格式需求文档直接应用于软件开发的过程中。因此，软件开发人员应当提炼原始需求文档的信息，根据RUCM模板，使之成为结构化的需求，将非形式化的需求转化为形式化的需求，撰写易于理解、无歧义的需求文档。通过本项目所设计的软件，软件开发人员将初始文档直接转化为RUCM格式的需求文档，从而提升效率，使开发流程更为规范。

针对上述目标，本软件应当在满足基本的文档转换功能的基础上，尽量保证较高的准确率与较快的处理速度。由于本软件的目标是代替原来人工分析需求、将之书写成相关格式的需求文档，故而准确率和效率就显得尤为重要。若提炼信息的准确率较低，则在需求分析这一步骤中，无法分析出客户的真正需求，或是误解了客户的需求，从而偏离了正确的软件开发方向，差之毫厘，谬以千里。若是文档处理效率低下，时间过长，则导致整个软件开发流程的时间的延长，与设计此系统的目的——提高效率、自动化处理背道而驰。因此，结构化需求文档处理软件在接受Given-When-Then格式需求文档的输入后，需要迅速对其进行处理，采用机器学习的方法自动分析需求信息，最终生成与原始文档中需求一致的RUCM需求文档。