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两个大类。其中，文档可以粗略地分化为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，人员可以粗略地分化为需求提出方、产品经理、开发人员以及系统支持四类。在下面，本文将按照以上分类分别进行描述：

1. 文档
   1. 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。
   2. 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。
         1. Steps（numbered）：事件流步骤
         2. Post Condition：基本流的后置条件
      9. Specific Alternative Flows：特定分支流
         1. RFS：引用基本流程步骤的名字与步骤号，例如RFS 3
         2. Steps（numbered）：事件流步骤
         3. Post Condition：分支流的后置条件
      10. Global Alternative Flows：全局分支流
          1. Guard Condition：进入该全局分支测试流程执行的守护条件
          2. Steps（numbered）：事件流步骤
          3. Post Condition：分支流的后置条件
      11. Bounded Alternative Flows（Test Sequence）：
          1. RFS：引用步骤的所在事件流的名字与步骤号
          2. Steps（numbered）：事件流步骤
          3. 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
   1. 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，其的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。
   2. 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。
2. 人员
   1. 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。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文档的了解深度不同可分为三类：
      1. 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。
      2. 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。
      3. 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。
   2. 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。
   3. 开发人员：此为RUCM文档的另一位需求者，他需要按照RUCM对于项目进行编码工作，同时通过RUCM文档对于项目整体的工作和接口有一个简单地了解。所以，在这里也需要保障RUCM文档的准确性和易读性，以方便开发人员的工作。
   4. 系统支持：此为系统的支持者，负责保障本系统准确高效的运行。本系统通过机器学习方法将GWT文档转化为RUCM文档。其生成的结果和构建的模型在有些时候可能不够完善，所以可以通过二次构建和修复来完善系统的稳定性和准确性，从而对系统进行迭代和进化。

一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，图一是用例图，图二是信息交互图。

图片包含 文字

已生成极高可信度的说明

系统用例图

GWT2RUCM

需求提出方

产品经理

系统支持

开发人员

GWT文档

RUCM文档

RUCM文档

标签与输出格式

模型优化参数

模型运行状态

信息交互图