



---

## WORK LOG 7

---

**Shandong University**

**April 19, 2020**

高德琛



# Contents

1	DFD 图 . . . . .	2
1.1	定义 . . . . .	2
1.2	$\mu Vlogger$ 数据流程图 . . . . .	2
2	OCL 语言 . . . . .	2
2.1	定义 . . . . .	2
2.2	优势 . . . . .	3
2.3	应用场景 . . . . .	3
2.4	$\mu Vlogger$ 逻辑流程图 . . . . .	4
3	用户文档和培训 . . . . .	4

## 1 DFD 图

### 1.1 定义

一图胜千言。数据流图 (DFD) 是系统内信息流的传统视觉表示。一个整齐而清晰的 DFD 可以用图形描绘出大量的系统需求。它可以是手动的, 自动的或两者的组合。

它显示了信息如何进入和离开系统, 什么改变了信息以及信息的存储位置。DFD 的目的是显示整个系统的范围和界限。它可能被用作系统分析员和任何参与系统的人之间的交流工具, 作为重新设计系统的起点。

它通常以上下文图开始, 作为 DFD 图的 0 级, 这是整个系统的简单表示。为了进一步阐述, 我们深入到一级图, 将较低级别的功能从系统的主要功能中分解出来。当需要进一步分析时, 这可能会继续发展成为二级图。级别 3, 级别 4 等等是可能的, 但是超出级别 3 的任何东西都不是很常见。请记住, 分解特定函数的细节水平实际上取决于函数的复杂性。

### 1.2 $\mu$ Vlogger 数据流程图

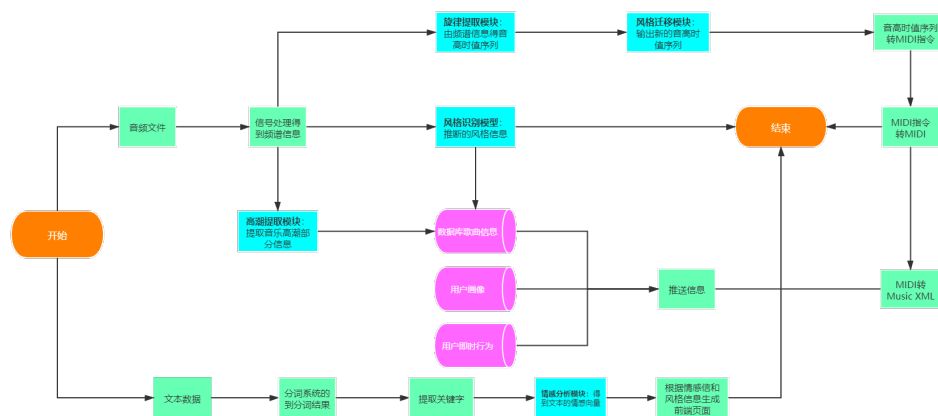


图 1: 数据流程图

## 2 OCL 语言

### 2.1 定义

一种用于描述 UML 模型表达式的形式语言。这种表达式通常规定所建模系统所必须满足的不变式条件和模型中所描述对象的检索。注意, 当计算 OCL 表达式时, 它们

不能有副作用（例如，它们的计算不能改变相应执行系统的状态）。

OCL 表达式可以用于规定操作 (operations) / 动作 (actions)，这些客体在计算时是改变系统状态的。UML 建模者可以使用 OCL 来规定模型中应用特定的约束。还可以使用 OCL 来规定 UML 模型上的检索，而这完全是编程语言无关的。

## 2.2 优势

UML 图，例如一个类图，通常不能精化到可以提供一个规约的所有相关方面。也就是说模型中还需要描述一些额外的对象约束。这些约束通常用自然语言来描述。实践证明这总是会带来歧义。为了编写无歧义的约束，形式语言应运而生。传统形式语言的弊端是它们只适用于具有很强的数学背景的用户，而对一般的业务或系统建模者来说很难使用。

OCL 被开发用来弥补这个鸿沟。它是一种形式语言同时易于读写。它作为一种业务建模语言诞生于 IBM 保险部门，根植于 Syntropy 方法。

OCL 是一种纯规约语言；因此，保障了 OCL 表达式是无副作用的。当计算一个 OCL 表达式时，只是简单的返回一个值。它不能改变模型中的任何东西。这意味着系统中的状态在计算 OCL 表达式时永远不会发生改变，即使一个 OCL 表达式可以用来规定一个状态改变（例如，在后置条件中）。

OCL 不是编程语言；因此，在 OCL 中不能编写程序逻辑或流控制。你不能在 OCL 中调用过程和激活非检索操作。由于 OCL 天生是一门建模语言，OCL 表达式是不能直接执行的。

OCL 是一种类型语言，因此每个 OCL 表达式都有一个类型。为了保障良构，OCL 表达式必须遵循该语言的类型一致性规则。举个例子，你不能拿一个整型与一个字符串进行比较。UML 模型中定义每个分类符代表了一个不同的 OCL 类型。此外，OCL 还包括了一些预定义的补充类型。

作为一种规约语言，它的实现超出 OCL 的范围而且也不能在 OCL 中进行表示。OCL 表达式的计算是瞬时完成了。这意味着在计算过程中模型中对象的状态不能改变。

## 2.3 应用场景

OCL 可以用于许多不同目的：

- 作为一种检索语言；
- 规定类模型中类和类型的不变式；
- 规定 Stereotype 的类型不变式；

- 描述操作和方法的前置-和后置条件;
- 描述 Guards;
- 规定消息和动作的目标;
- 规定操作上的约束;
- 规定 UML 中任意表达式中属性的衍生规则。

## 2.4 $\mu Vlogger$ 逻辑流程图

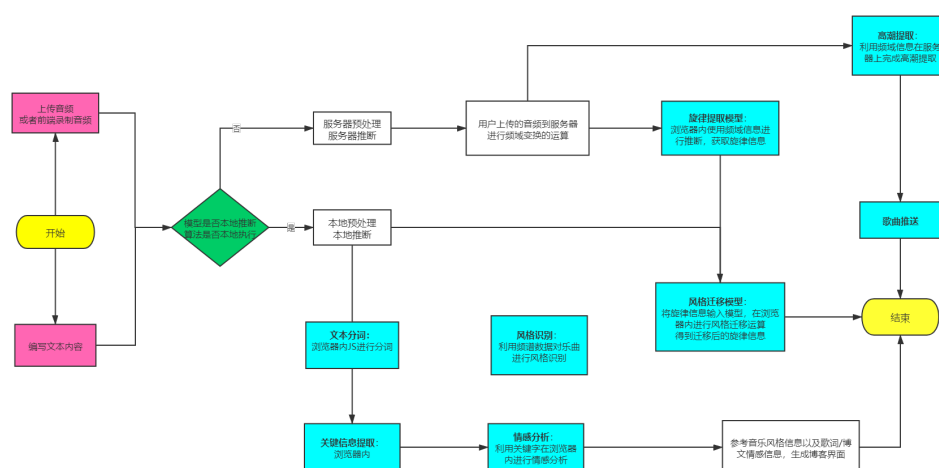


图 2: 逻辑流程图

## 3 用户文档和培训

1. 为了  $\mu Vlogger$  完成后提供给用户一个安装、操作使用说明, 提高售后人员的响应, 帮助用户更好的操作使用产品。而在另一方面, 面向小组内部销售、研发、设计等人员, 提供给销售理解产品, 提供给研发作为开发依据, 提供给设计作为设计框架。
2. 除此之外, 用户文档还要主要介绍  $\mu Vlogger$  主要内容与功能。通过对每个功能进行详细的描述, 可以让阅读者快速全面地了解该系统的功能。方便客户和需求分析人员之间的沟通, 让设计人员、测试人员、开发人员更好的了解用户的需求。

### 3. 文档撰写框架比较软件产品用户手册

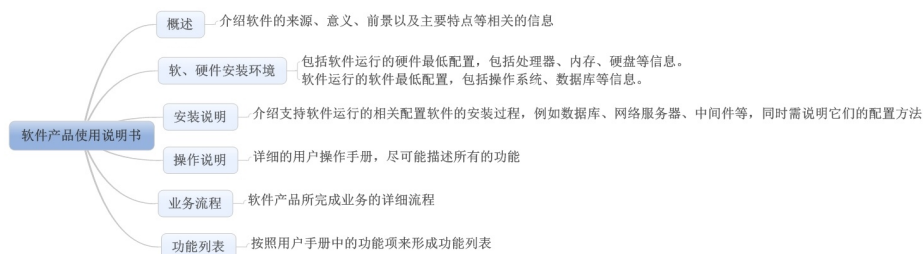


图 3: 软件产品用户手册

### 4. 软件需求文档

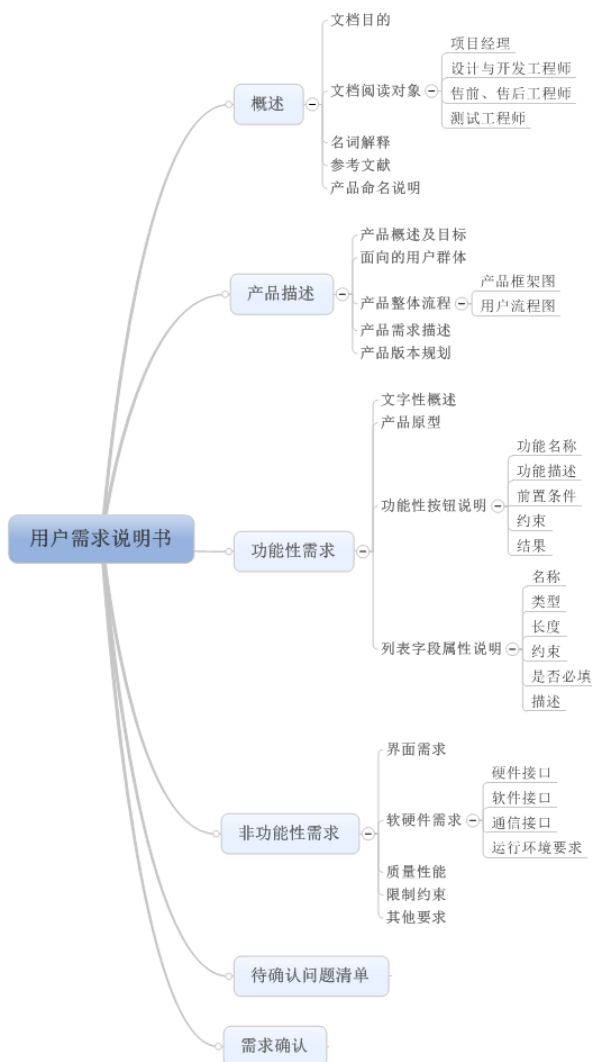


图 4: 用户需求说明书

5. 总结：软件产品使用说明书的背景可参考用户需求文档中产品描述中的产品概述及目标、面向的用户群体；软件产品使用说明书中的业务流程和用户需求文档中的用户流程图基本一致，可根据需要进行相关的扩展。软件产品使用说明书中的操作说明为重点，可结合用户需求文档中的功能性需求说明来尽可能详细的说明软件操作的一般过程。