软件工程形式化方法论文总结（一）

16121731 皮怀雨

October 9, 2016

1 Woodcock & Davies Using Z

通过阅读Using Z Introduction章节，对于形式化方法的作用、Z的特点和证明的重要性有了一定的理解。文章中还有两个例子，分别讲了IBM的CICS系统和伦敦地铁图的改进，让我对形式化方法有了更加直观的认识。

随着软件的广泛使用，系统变得越来越庞大，文档的复杂性也与日俱增。所以，文档中的关键问题往往会因为描述不到位或者开发人员理解有误被忽略，而由此造成的错误修改起来难度和代价非常大，有些错误甚至给客户造成严重的损失和灾难。我们应该掌握一些技巧去尽量避免这些错误的发生。通过使用形式化的方法描述系统，编写的文档不仅具有简洁和清晰的优点，而且能够在设计、开发、测试和维护各个软件工程过程中使用。我们使用形式化方法编写文档的主要问题在于，开发团队是否能接受另一种模式的开发，也就是他们是否能够适应通过一套规范的语言和数学符号来表示整个系统。在CICS系统的工程实践中，通过一定的训练，没有数学基础的和特殊符号使用经验的开发人员能够很好的完成这项工作。所以，通过形式化方法来设计软件是可行的。

Z的表示是基于集合论和数学逻辑的。它有很多良好的特性，不仅易于学习，还是可以结构化的。通过schemas，数学对象和属性可以被很好的表示和描述，通过这些schemas，我们可以描述整个系统。Z的一个特性是类型化的，每一个在数学模型中的对象都被指定了固定的类型，固定类型的好处是便于之后使用相关的算法进行检测。还有一个特性就是自然语言化的，可以通过自然语言命名不同的对象和属性，是的系统具有良好的可读性。Z的模型还可以不断地细化和改进。如果需求发生变化我们可以对模型进行相应的修改，还可以对模型反复的精炼，提高模型的正确率。这样的修改可以一直持续到模型自动翻译成可执行代码之前。证明在形式化方法中是至关重要的，它可以提升软件质量并具有可行性。在证明的过程中我们可以更好的理解用户对软件的需求，防止因为需求不明造成的错误。在设计的过程中，我们通过证明，来确认现阶段的设计是否合理，它为什么是正确的。对证明的训练也更加有利于我们构建正确的形式化模型。

通过最后伦敦地铁的例子理解，我认为一个优秀的形式化表示应该是抽象、简洁、完整、可维护和对用户友好的。通过形式化的表示，让一个系统变得更加简洁和精炼并且易于理解，这是形式化方法的突出优点和重要作用。形式化方法的表示，使得软件项目的文档变得严格可证明，而且它代表了软件系统的全部结构。这样的方法极大的降低了软件系统的错误率，提高了可维护性。

2 工业开发中的形式化方法：成就、问题和未来

通过对Jean-Raymond Abrial的报告“工业开发中形式化方法”的阅读，了解了一种新的软件工程开

发方式，从中也理解了一种更加严谨的工程方法。它对于传统的开发方式有了一个全新的颠覆，有着更加严谨的规范。

在软件开发日益复杂的今天，软件的稳定性、安全性和复用性越来越重要。但是在常规的软件开发中，往往对于需求分析和建模投入的资源有限，主要通过开发过程的控制和后续测试的反复，来提高软件的稳定性和安全性，但是这种方式往往存在着很多的安全隐患。在软件开发的过程中和测试后对系统进行修改和更新付出了大量的时间精力，代价非常大。对于这种模式，我们应该进行一些反思，是否能使用更加严谨的和可控的工程方法控制软件的品质？在成熟的工程领域，往往可以通过数学的计算和模型来验证工程是否达到各项指标。

在很多对安全性要求较高的领域，比如轨道交通和航天，采用更加可靠的开发方法非常必要。如果采用形式化方法对软件进行需求分析、抽象模型然后转化为具体模型和自动生成代码的方法开发，就会得到相对正确的代码（对于需求和模型）。形式化开发的过程主要分为3个步骤，首先是进行需求分析，这是所有软件工程的必要阶段，而且非常重要。准确的需求分析决定了软件最终是否符合用户的要求，在有了准确的需求之后，形式化方法和常规的开发有所方法不同。然后就是进行逐步求精建立抽象模型，在这个过程中，需要大量的人工参与，构建好抽象模型是形式化方法中耗费最大的步骤。对比常规的开发方法，虽然也需要构建模型，画出用例图、流程图和类图等为后来开发做参考，但是在这个过程中耗费的人力在整个软件工程中是较少的。从得到的抽象模型，我们可以得到具体模型，这个过程不再不要需求文档，而且这个过程的人工参与不如之前的广泛。最后一步是模型自动翻译成为可执行代码，这一阶段不需要人工的参与。而在常规的软件工程过程中，编码往往是最为耗时的，而且会出现很多错误。但是，自动翻译也是现在形式化方法很大的挑战，如何自动产生高质量的可执行代码还是一个难题。

按照软件工程形式化方法，软件工程师通过需求分析和逐步求精建立模型，可以最终翻译成为一个安全和可复用的系统。如果软件工程师只要完成相关的模型构建，那么就不需要他们去具体的编写代码，这样降低了软件开发过程中对不同新技术的掌握成本。软件工程师只需要掌握形式化方法，就可以构建正确的软件。使用形式化方法构建软件系统，省去了软件开发繁重的编码过程，而且降低了系统的错误。