**Abstract of *Taming Uncertainty in Self-Adaptive Software***

周昊一 mf1233055

自适应软件指的是能根据外部环境的变化自动的调整自己的行为，以完成目标的软件。在自适应软件的运行中会遇到一些不确定的问题，这些问题会影响软件的自适应过程，导致软件行为发生异常。不确定的问题可以分为两类，一类是由外部环境改变的不确定引起的，如天气的变化。另一类是由内部环境变化的不确定引起的，如切换到某个模式是否能达到省电的目标。本文的目标是减少后一类的不确定情况带来的影响。

这一类的不确定性所带来的不利影响是由于软件的自适应行为都是基于实现定制的规则实现的。而这些规则依赖于外部环境的变化，通常是某个或某些参数达到某个值，就会触发某个转换规则。但是在不确定的情况下，不能保证系统的状态就处在规则所设定的参数上，比如环境的参数不一定是像所设想的一样在一个枚举集合中变动，而是会取一些浮动的值；再比如软件所采用的分析模型在对系统的一些参数进行量化时，可能采用了简化的方法，忽略了一些参数，这就造成了最终结果可能与实际情况有偏差；还有可能用户的所设定偏好与他实际的偏好之间也是有偏差的。这样也就不能保证按照这些规则能够达到最良好的自适应效果。

文章介绍了一种叫做概率自适应（POISED, POssIbilistic SElf-aDaptation）的方法来处理在有内部不确定性的情况下，自动进行适应性决策时的复杂情况。这个方法将系统的自适应用概率的情况来考虑，当外部环境发生改变时，并不是按照固定的规则进行状态的转换，而是采取在这种情况下最可能满足目标的行为。POISED对不确定情况的分析处理基于两个能力：一是分辨不确定性的来源；二是估计不确定性的水平。因此还要涉及到对不确定性进行量化的工作。最后还要将系统的目标进行量化，以根据概率方程来解出当前状况下所应该采取的行为，这里主要涉及到一些数学方面的计算。

随后本文用一个现有系统作为例子，对比原有方法和POISED，对POISED的效果进行了评估。评估主要从以下几个方面进行：一是选择的结果的质量，即比起原有方法是否能做出更正确的选择；二是资源的消耗，即在有限的内存等环境下是否能顺利运行；三是在多个目标权重不同的情况下是否能正常工作；四是对不确定情况估计的敏感性；五是最终系统的运行效率。