“解决问题”与“消灭问题”

一直以来，人们都重视“解决问题”的能力，却忽视了另一种重要的能力：“消灭问题”的能力。各种各样的竞赛，分数和排名，让很多人从小就片面的认为，能“解决问题”的人，就是最厉害的人。拿到一个问题就埋头求解，很少考虑这问题到底有什么意义。这种呆板的思维方式，不仅存在于低级的“应试”和“解题”过程，而且蔓延到了很多艰深的研究领域。

如果你仔细观察就会发现，很多“难题”，其实是“人造”出来的，而不是“必然”的。它们的存在，往往是由于一些早期的“设计错误”。人造的东西里面往往有设计上的错误，如果你把这些东西看成是不可改变的东西，那你就会遇到很多不必要的问题。打个比方，如果当初轮子被设计成方形的，而没有人质疑这样做的“必要性”，那么也许人类早就因为“能源问题”而灭绝了。有点夸张，但它却形象的说明了，为什么错误的设计会导致不必要的难题。

其实如果我们转换一下思路，或者改变一下“设计”，很多问题就可以不解自消。这就是我所谓的“消灭问题”的能力。这种“消灭问题”的能力，表面上容易其实难，有点像脑经急转弯，所以经常受到人们的忽视。看到一个问题轻而易举的消失了，总有人满不在乎的说：“这个容易。我也能做到。” 可问题就在于，你怎么没想到？说这种话的人，完全没有意识到，他们的思维里面其实缺少了非常重要的东西。由于喜欢炫耀自己的“头脑暴力”，他们经常解决（甚至制造）错误的问题。

所以，在解决问题之前，我们应该先问自己三个问题：

1. 这问题是否真的“存在”？

   也许你已经看出来了，很多问题，即使众人都认为它存在，其实也可能是不存在的。在这一点上不能相信任何人或者机构，不管他有多么的“权威”。就像小马过河的道理，只有靠自己的实践。

2. 如果解决了这个问题，会给我和他人带来什么实际的好处？

   世界上不存在“永远”，也不存在“无穷”。如果一个“科学算命家”花100年才能算出我的未来，那我还不如坐等“未来”的到来。所有的人，都不过是来这世界上做短暂的旅行。所以，问题的答案，应该能在合理的时间之内给人带来实际的好处。

3. 这问题是否可以在简单的改变某些“设计”或者“思路”之后，不复存在？

   很多问题的“存在”，其实是因为人们的“思维定势”。他们看不到问题的“根源”和因果关系，而是经常在下意识里假定某种“先决条件”（A）的存在，然后坚定不移的相信由此“导致”的问题（B）的存在，如下图：

```
A -----> B
```

然后，他们开始呆头呆脑的解决 B，完全忘记了质疑 A 存在的必要性。他们从来没有想过，如何消除 A，或者切断 A 与 B 之间的关系。他们没有发现，一旦这前提 A 不复存在，问题 B 就可以不解自消。

对这一点，我想起一个有趣的故事。有人在饭桌上给大家出了一道“难题”，要他们把自己盘子里的鸡蛋立起来，最后只有一个人做到了。这个人把蛋壳打破了。所有其他人都没有想到这个做法，却说他“犯规”。可是应该检讨的其实应该是他们自己，因为出题的人根本没有说不能打破蛋壳，他们却对此做出了错误的假设。

我经常发现计算机科学界存在这样的问题。研究了几十年，结果到最后才发现，辛辛苦苦解决的问题，其实包含了错误的假设。如果换一个角度来看，或者稍微改一改设计，这问题就基本不存在了。其中一个例子，就是编译器里面的“语法分析”（parsing）问题。

语法分析成为一个问题的原因，就在于很多人错误的以为程序语言应该有复杂的语法。正是这些复杂的语法，造成了这个问题研究了很多年，仍然没有一个很好的解决方案。可是一旦语法设计被简化（比如像 Lisp 那样），语法分析就变成一个非常容易的问题。实际上计算机系统（比如 Unix）里的很多问题都是由此引发的，想要利用字符串来进行数据交换，却又设计了一些非常不方便的“数据格式”。简单的语法设计，会让这些问题一并消失掉。关于这个问题，我不想重复发文，细节请见另一篇博文《谈语法)》。

爱因斯坦说“想象力比知识更重要”，也许就是这个道理。没有想象力的人经常钻牛角尖，走死胡同，忘记了自己其实还有另外的路可走。