

TURING

图灵原创

大道至简

实践者的思想

周爱民 著



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵原创

大道至简

实践者的思想

周爱民 著

人民邮电出版社

北京

图书在版编目 (C I P) 数据

大道至易：实践者的思想 / 周爱民著. -- 北京：
人民邮电出版社，2012.6
(图灵原创)
ISBN 978-7-115-28217-0

I. ①大… II. ①周… III. ①软件开发—项目管理
IV. ①TP311.52

中国版本图书馆CIP数据核字(2012)第103852号

内 容 提 要

本书可以看成是《大道至简》的姊妹篇，是以软件工程为体系、以组织结构为视角，融合系统架构师、项目管理者和软件开发人员三种角色实践的思想总成。本书讨论这些思想，并陈述它们所基于的原则、背景与获得过程。

本书适合各类工程管理人员、软件开发人员和架构师阅读。

图灵原创

大道至易——实践者的思想

-
- ◆ 著 周爱民
责任编辑 杨海玲 李松峰
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本：800×1000 1/16
印张：25
字数：519千字
印数：1—5 000册
- 2012年6月第1版
2012年6月北京第1次印刷

ISBN 978-7-115-28217-0

定价：69.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223

反盗版热线：(010)67171154

与架构师同行（序）

我认识周爱民多年，本书是他的第五部作品，也是“周爱民风格”最浓烈的一部。作者把“冥想”式的写作发挥到了新的高度，任何一个小小的、普通人熟视无睹的问题、现象、矛盾，都能引起作者滔滔江水连绵不绝的思考，追根溯源的论辩和富有禅机的跳跃的联想。这是周爱民思维的特点所在，他能够长时间连续不断地思考一个问题，越挖越深，既不知疲倦，也不向现实妥协。这是一种独特的思维品质，是如我一样普通人所难以企及的，也是这个时代所稀缺的。这种独特的思维品质，既是这本书价值的源泉，也成为对读者，以及我这个受托作序的非技术人来说最大的挑战。所以在这篇序言里，我不敢说能把握这本书的思想实质，但会坦率地谈谈我对书中所涉及的一些问题的个人看法。

首先是关于这本书的主题和内容。这本书当然是在谈软件开发的问题，但涉及的方面又不仅仅是开发，甚至可以说，作者主要不是在谈开发，而是在谈组织。而且他不是谈某一种具体的组织，不是代码的组织、构件的组织、架构的组织，甚至也不是工程的组织和人的组织，他是在整体上来思考“组织”这件事情的一般原则，然后用其所思所得来“格”代码、“格”架构、“格”工程、“格”人的组织。从书中，读者可以跟着他的思路去体会他是如何“格物致知”，然后又反过来“致知格物”。

为什么要把组织问题放在中心位置去探讨？因为良好的组织是解决软件规模问题的唯一希望。而随着软件行业的逐渐成熟，组织和架构已经取代编码而成为软件开发的核心问题。

十几年来软件开发行业发生了一个重大的变化，那就是软件开发已经渐渐演变成了一个以配置为主、编程为辅的过程。这个观点最早是 20 世纪 90 年代中期由著名的 C++ 网络编程框架 ACE 的创造者 Doug Schmidt 教授提出来的。当时一系列重要的基础性的标准，如 C 语言、TCP/IP 协议、POSIX 标准已经奠定，Schmidt 教授意识到，

一旦一小部分人开发出足够多的、高质量的、可配置的核心软件，其他人就没有必要重复发明轮子，而是可以站在他们的肩膀上，通过配置和扩展来完成任务。尽管配置的过程本身还是通过编码的方式来完成的，但本质上这些代码只是在定制核心软件的运作方式，它们的技术难度比编写高质量核心软件至少要低一个数量级。

这个观点被历史证明是非常具有远见的，特别是随着 Java 等跨平台语言的风行、开源运动的兴起和 Web 的成功，绝大多数不同种类的应用都在共享相同的基本特征，这样也就出现了一大批门类齐全、质量过硬的核心知识资产，它们以基础软件、中间件、框架、程序库和开源代码的方式存在，可以被以各种方式复用。在这些资产之上，程序员社群形成了有效的、开放的知识分享机制，从而使得一般性的软件开发变得相对容易，而将软件开发这件事情的挑战上移到架构层面，或者更具体地说，如何有效地分配职责、组织资源、建立资源节点之间的关系和契约，已经成为大规模软件开发的根本问题。

这就是本书的发力点。作者正是站在架构师的高度上来看待软件开发，甚至更大的意义上，看待以软件开发能力为核心的企业在产品战略、人力配置和文化方面的组织问题。作者在过去几年里，先后在两家超大规模互联网企业中做架构师，他在其中的矛盾、迷惘、挣扎和成就是这本书的思想源泉，而他对于架构师角色的实践和领悟，也是这本书的价值所在。作者认为，软件开发发展到今天这个阶段，架构问题是真正的方向性问题，架构之争是真正的路线斗争，架构错误是唯一致命的、不可挽回的错误。只要架构适用，其他的错误都是局部性的、可以控制的。一旦架构发生方向性错误，无论是人的架构，还是软件的架构，其代价可能就是一个项目甚至一个企业组织的生命。因此，这本书通篇都在讲架构问题，讲组织问题，讲规模问题。读这本书的时候，如果能够始终记住这一点，那么理解起来会容易一些。

看上去这本书的四个篇章，彼此相对独立，谁也不挨着谁。但实际上，贯穿全书的主轴，就是从宏观的组织关系上和微观的角色职责上，在不同的领域里反复谈组织架构问题。在这个过程中，作者也提出了一些鲜明而有趣的想法，比如让架构师成为开发团队中的核心角色，甚至把人员和项目管理这样传统强势的角色也仅仅视为架构师的支撑角色。作者的核心观点，说到底很朴素，无论是在人员组织还是技术架构上，作者提倡各就各位、各司其责、不增不减、不垢不净的风格，尽量简化由于异动而导致的非本质复杂度。这些观点，至少成一家之言，值得思考。

当然，用这种“中心思想”、“段落大意”式的语文课方式来归纳这本书，是不合适的，因为这本书的写作风格确实是非常独特。以周爱民的水平和资历，他大可以像很多流行书作者一样，居高临下，弄出几条十几条原则，以一种非常具有“架构感”的组织方式来组织这本书的内容本身。但是如果架构问题能够以这种方式讲清楚的话，那它早就不是问题了。如作者所说，架构本身是“以序容易”，是为了约束系统自由度，从而使架构之外的决策空间更小、更容易，那么架构本身的设计，则成为

重点和难点。如何成为好的架构师，如何教授和交流架构设计的经验和思想，迄今仍然不能说找到了办法。作者采取的方式，就是老老实实把自己的思维过程和喃喃自语都很原生态地记录下来，从而形成一本“心语”之作。因此，如果读者也能够“心心相印”，潜下心来仔细品读这本书，那么就相当于跟作者一起进行了一段旅行对话，或者，一起在思维的教武场里杀个几进几出。或许这种如切如磋、如琢如磨、反复捶打、反复锻造的风格，倒确实能够帮助有心的读者获得一些架构设计的经验。

因此，无论从内容上，还是从行文风格上，这本书都是非常独特的，也有其独特的价值。作者的作品，一向是长销之作，其价值也往往需要一段时间才能被真正认识，我不敢说自己能够在短暂的阅读当中充分理解这本书，但是我相信其中所蕴藏的价值和诚意。

当然，无论对于书的观点、内容还是展现形式，我也有自己的看法，其中也有不同意见。

比如在行文上，我认为作者对于读者设置了比较高的理解门槛，要求读者必须跟着他的文字和思绪流动。如果读者以当下流行的速读式阅读、跳跃式阅读甚至微博式阅读的方式来读这本书，我敢说不但不可能把握其中的精华，甚至连一些基本概念可能都会搞错。

再比如在内容的铺述方面，上至人员组织架构，下至程序设计语言中一些架构性的设计，在一本 300 多页的书里跨越如此巨大的领域，我相信是前无古人。尽管作者确实始终以一主线贯穿，但是能够洞解其中滋味的读者，恐怕也是少之又少。好在读一本书，也并不是要求全部理解，懂一点，学一点，常读常新，也是一种不错的体验。

至于作者的思想，我能够置喙之处不多。勉强要说一点，可能就是一个搭配问题。从组织、人员、文化到技术、编码，既然每个层面上都有自己的架构问题，都有若干不同的架构选择，那么可能不同层次之间架构选择的搭配问题，或许也应该是这本书应该触及的。一个优秀团队的人员组织架构，必须与其行业领域和产品项目类型彼此搭配，也必须与其选择的技术框架和架构彼此搭配，也必须与其软件过程的管理方法彼此搭配。抽象地在每一个层面上去谈什么是好、什么是坏，是没有意义的。CMMI 的模型是个好的过程架构，但是谷歌和 Facebook 决不会采纳，因为这个架构与其他领域的既定架构不兼容。什么样的组织架构与什么样的技术架构搭配是合适的，这本身是一个非常有趣和有意义的问题，可惜这本书落墨不多，不得不说是个可以改进的地方。

总的来说，这是一本需要认真读的书，也是一本值得认真读的书。它的价值需要读者与作者在思维互动中逐渐体现。如果你是这样的读者，我会愿意向你推荐这本书。而如果你并不喜欢深彻的冥想式的阅读和思考，那么你很难从这本书中获得太多。

孟岩

写给读者

这不是一本能够速读的书。

如果您读过《大道至简》，那么本书会完全颠覆您此前的阅读体验。尽管它们可以看成姊妹篇，并且在内容上也有承接，但却有着完全不同的写作方法与思维系统。又如果您了解传统含义上的“软件工程”，那么也请您先放弃固有的观念。本书所述的，在您理解之后尽可以同样放弃：所有接近与远离真理的，都必被打破。

这是一本面向多类读者的书。

本书是“实践者的思想”，并没有确指是“软件工程实践”。因而所言的是以软件工程为体系、以组织结构为视角的，在系统架构、项目管理和软件开发三种角色实践中的思想之总成。如果读者是其中的某一角色，建议从相应的分篇入手阅读；如果读者同时兼有多种角色，并有丰富的实践经验，可以从总论入手阅读。

这是一本讨论思想方法的书。

如同《大道至简》一样，本书不讨论具体的做事方法。所有行事的方法与准则，都不能孤立于其背景而存在，因此方法是死的，得到这些方法的思想是活的。但亦如行事的方法一般，思想的方法也不能脱离其背景存在。所以本书讨论这些思想，并陈述它们所基于的原则、背景与获得的过程。

这是一个视角而非全景。

无论如何，我们只能触及事物整体的一个或几个侧像。以管窥豹并不可笑，可笑的是每个人都认为他看到了豹的全部。认识到我们自己的眼界之狭隘，比得到这个狭隘认识中一个确实的影像更为重要，前者是进步之始，后者则是自满之源了。本书所讨论的工程模型仅是软件工程的一个视角，只是我将其当成软件工程的本质问题来讨论，毕竟无论何种工程，本质上仍然是人的行为，而不是事的流程。

前言

易是变化

台湾的高焕堂先生曾说架构的要旨是“以序容易”，我解释成“用规则来包容变化”，高老师说很合他的本意。这里的“易”，指的就是变化。既是变化，那当然是艰难而复杂的了。然而我们通常说一件事易做或另一件事不易做，这里的“易”却都是指简单的意思。所以“易”既是无穷的复杂，也是至极的简单，关键在于如何“容”它。

我们得看清什么是变化。

盲人摸象是一个很好的故事，因为每一个盲人的认识都是其固见的、自见的，以及自证的真理。然而盲人们的观点放在一起的时候，却是一个笑话。因为看笑话的人看到了“变化”，而这些局外者原本认为那头大象是不变的、确定的、唯只有一个形象的。同样，面对任何我们所见的变化，或对变化中的任何一个认识，我们都认为那不过是笑话。再深彻地透视这一点，其根本在于：我们也有一个对大象的“认识”，只因为这一认识看起来——或我们认为——比那些盲人更为高明、正确，所以我们才别出了盲人，看到了笑话。

正是我们对一个事物固有的认识，制造了盲人与盲人的笑话。

倘若我们的认识是不可易变的，那么我们今天就已经看到了真理，看透了世事万象，我们已成至人，故而我们不需要存在亦无需进化：一切于我们而言，必须静止；一切于我们的认识而言，不可复加；一切于我们的思维而言，不可偏侧。而这，看起来不正是我们自盲了双目，自演了笑话吗？

反过来，我们得承认变化的存在。我们所要做的，只是承认自己是一个盲人，可以从一个方面去触及这一事物，形成一个认识，用一些规则、规格或概念去确指它。

用同样的方法，我们触及这个事物的方方面面，进而得到与这个事物最为接近的一个全像，这就是我们关于这一事物所有的、而又未尽的知识。

知音变而得律，有容则易。

简也不是简单

这本书是《大道至简》（以下称“《简》”）的续作，那我又是怎么认识《简》的呢？究我们的认识的根底来说，我们向来不是缺乏思考的。我们有着种种在实践中的选择，这些选择是基于一些认识的，而又产生了一些新的认识。而我所有的醒觉，并不来源于对后一种认识的反复持续，反而在于对前一种——既有的认识的观察。

《简》一书的源起，正是一次对“我在干什么”的发问。在这一反思中，我最终找到了我在工程中的种种位置，并将这些位置在我既有的知识中——标点。然后，我们就看到了那幅 EHM（工程层状模型）图（参见“总论”图 0-1）。

追究起来，EHM 图以及它所代表的我的认识已经是 8 年前的事情了；完成《简》一书，也已经是 6 年前的事情了。6 年前，我将《简》作为电子版发布的时候，也同时得到了一个新的职务：架构师。然而如果你仔细地观察 EHM 图，就会发现它根本上是不能容纳“架构”这个角色与职责的。

“我在干什么”，仍然是我这 6 年来最主要的思考。事实上，这也促使我抬起头来，观察周围营营碌碌的人们，了解他们在做什么，在围绕什么去做，以及——也是最重要的是：他们得到了什么。如同《简》中所说的愚公一家，数百年挖山不止，最终却不得不筑关自守，我们是不是也一样背离了初衷，成为一个不知所为的愚者？

《简》事实上是打开了我思维的一扇门，让我看到“我既有的那些东西”是何等粗糙，又曾经是如何粗暴地塞入了我的头脑。例如我在计算机专业中所学的第一门功课是《数据结构》^①，这本书的第一页便写着：“用计算机解决具体问题，首先要加以抽象。”但十余年来，我从未思考过“什么是抽象”，又或者我的思维以及由此驱动着的软件开发活动中“何处是抽象”。抽象被作为我们——程序员的一种事实的本能，或被驯育的技能塞入了我们的举手投足，控制着我们而又让我们无知无觉。

《简》在这“无知无觉”的宁静中轻一叩击，“叮”的一声，我的眼前便亮了。观之简，便在其原始的粗鄙；思之得，便在行道中的一凛。

没有答案

《简》一书只提出了现象，而无有答案。2009 年在北京的 QCon 大会中，我跟几

^①《数据结构》，严蔚敏、吴伟民编著。

位朋友谈到过这个问题。我坦承这一事实：若《简》尚有一点可读处，便在于它粗暴而又挑衅地把种种问题摆在我们的面前。但对许多人来说，这毫无益处。因为他们或是已经看了这些，或已经为之麻木，或是在贪求某种法子来逃避这些东西带来的影响。

总而言之，《简》是“没有答案”的。我彼时也是“没有答案”的，我只是觉醒过来，并看到了一些事实上存在已久的东西而已。与此相同，《大道至易》这本书也“没有答案”。尽管我提出了一些貌似答案的东西，但我确实是在保守地、谨慎地谈论它们，并且时时地（譬如现在）警示读者：这是在某些背景下的一时之选。

“讲述某些我知道或想到的答案”是本书中最没有价值的部分，即使它们存在，即使看起来“非常有用”，也仅是一时的表象而已。

与《简》不同的是，《大道至易》提出了种种问题。就我的认识来说，现象与问题之间是有着一个巨大的跨越的。例如《简》中提到的“跑不动了”，它可以成为一种催人奋进的现象，也可以是脱水这样的问题。作为旁观者，若只见现象，便只会责骂鞭策；若能见问题，则知道“一杯冰水”才是解决之道。在我看来，正确的问题与“解”之间的距离是可显见的，而一个表面的现象或错误的问题，就离真正的解森远而无际了。

《大道至易》是一本提出问题的书。

不设节名

正因为问题重要，而答案不重要，所以我认为给每段内容加以一个“有诱惑力的标题”是多余的。因此本书分篇、章、节三级目录，但节下的小节只有“一、二……九”这样的中文标号，没有标题，是谓“不设节名”。

事实上在我撰写的过程中，这些小节多数是有标题的。但最终我发现，若是有一个标题，则读者便非常容易陷入这个标题的“含义”之中，以之作为小节之纲要、主旨或主张。而这又正是我所不愿意的，因为我原本只是拿这些标题来搭个架子而已。

也就是说，本书原本是只有骨架，而没有血肉的。真正有血肉的，是你面对它的种种思考：那些没有名字的小节，是可以由你去补个标题的；那些没有定论的观见，是可以由你去设定的。我要做的，只是把问题写出来，把我的想法写出来，把我的观见写出来。这些，既不是真理，也不是我自作了固见的认识。

我之于盲人，相距并不甚远，只是我惊觉我的盲目，故而对周围多做了些探索；又或者我还能觉出蒙眼的黑布才是真实的问题，揭去之后，便一切历历在目、观之若即了。

只不过一刹那，定是相当刺目的。

不为读者

对于许多读者而言，本书的“总论”是一个过高的阅读起点。因此本书的编审老师曾建议将“总论”移至最末一篇。为此我犹豫了很久，并也曾数次动摇。最终，我仍坚持将它作为本书的开篇。尽管编审在来信中写道：“根据我们的经验，好的开头对销量的影响可能达到三成。”

这真的是不小的诱惑。

相比来说，本书的第一篇“似乎”更适合作为“好的开头”——这也确实是编审老师们所推荐的。然而，事实上这一篇我是最晚成稿，写得最累，以及认为它是最不易读的。尽管表面来看，读者可以从这里轻松入手读上几页，然后觉得这是本“还有点趣”的书。然而它到底写的是什么呢？

离开总论的框架，所谓第一篇，不过是可以换得你的同感的抱怨而已。而真相，绝非如此。就写书出书而言，我当然可以用读者喜欢的方式去写作，并用读者喜欢的方式去包装那些篇章，然而这样一来，读者便会在“一开始”就失去整体的、系统性的认识，并认为这本书不过是又一本讲述软件工程的《大道至简》。

我可以为了“读者的口味”而改变这本书的著述体系吗？这是我最终的设问。而我很快地给出了答案：不！

总论是本书所述思想的核心，而后续几篇则是这一思想下的几个视角。让读者抛开总论而去窥见某个视角，与盲其目使之摸象何异？与予其管使之窥豹何异？我在前言中所反对的思想与方法，何以成了我去争取“读者人数”与迎合“读者口味”的手段？！

因此写下这段文字，以此自勉：莫因读者“买书如买纸”，便将书作了“字纸”去写，又或将书作了“纸页”去卖。

既如此，便仍让总论做了第一篇罢。这样一来，即便读者从它处读起，也总还能时时记得：有一篇“总论”在前头，或许蕴含着作者思想的主旨。

致谢

感谢所有的读者、编者以及一直以来支持我的朋友们。

感谢孟岩先生为本书作序。

感谢图灵出版公司。

感谢我的小学数学老师。

感谢 Joy. En，……我最爱的妻。

目 录

总论 领域角色的关注	1
第一节 什么是领域角色的关注	2
第二节 基于组织视角的观察	11
第三节 工程的本质问题是组织	25

第一篇 具体工程下的组织与行为

引言 管理中的逻辑	33
第一章 任人治事：组织行为的基本认知	35
第一节 刺秦与灭秦	36
第二节 看到别人能做什么	43
第三节 要做事，不要管理	51
第四节 伯夷与叔齐是怎么死的	59
第二章 谋定后动：项目的存在权	67
第一节 试错通常是无能的托辞	68
第二节 合法的山大王为什么没能成功	78
第三节 自己想办法	86
第三章 具体而微：工程是系统而不是事	93
第一节 做事的选择	94

第二节 你要什么	102
附录一 行在道上，从局部到全局	109
附录二 本来面目——大教堂、集市，与作坊	111
附录三 杀不死的人狼——我读《人月神话》	113

第二篇 程序源流：从计算到系统（上）

引言 简单的本源	117
第四章 计算系统	119
第一节 数，以及对数据的性质的思考	120
第二节 逻辑	126
第三节 抽象	129
第五章 语言及其面临的系统	133
第一节 语言	134
第二节 从功能到系统	141
第六章 程序设计的核心思想	151
第一节 数据结构：顺序存储	153
第二节 数据结构：散列存储	160
第三节 执行体及其执行过程中的环境	167
第四节 语法树及其执行过程	174
第五节 对象系统：表达、使用与模式	181

第三篇 程序源流：从计算到系统（下）

第七章 应用开发基础	199
第一节 应用开发的背景与成因	200
第二节 应用开发技术	207
第三节 开发视角下的工程问题	216

第四节 应用程序设计语言的复杂性	223
第八章 系统的基础部件	231
第一节 分布	232
第二节 依赖	241
第三节 消息	245
第四节 系统	250
第九章 系统的基本组织方法与原理	253
第一节 行为的组织及其抽象	254
第二节 领域间的组织	257
附录一 “主要编程范式”及其语言特性关系	273
附录二 继承与混合, 略谈系统的构建方式	275
附录三 像大师们一样思考——从“UML 何时死掉”谈起	277
附录四 VCL 已死, RAD 已死	279

第四篇 架构的思想与指导原则

引言 架构师的思维	285
第十章 你所关注的系统	287
第一节 了解系统的过程	288
第二节 知识的构建	295
第三节 最初的事实	307
第十一章 架构是过程, 而非结果	315
第一节 架构师的能力结构	316
第二节 系统架构与决策	324
第三节 架构的表达与逻辑	338
第十二章 架构原则, 技艺、艺术与美	355
第一节 架构原则	356

第二节 技艺、艺术与美	368
附录一 做人、做事，做架构师——架构师能力模型解析	373
附录二 专访：谈企业软件架构设计（节选）	375
参考资料	377
图表索引	382

总论

领域角色的关注

第一节 什么是领域角色的关注

屁股决定脑袋，这其实并没有什么不对。但首先不要关注屁股的大小，而要关注它的位置。在一个系统中，如果你都不知道自己坐在哪里，那么绝对不会知道自己该说什么、该做什么。位置的价值并不存在高下，也不可以拿某种方式去度量。尽管我们可以用收入或财产去衡量一个清洁工与一个老总的身价差异，但你无法去衡量“清洁一间厕所”与“看一个小时报告”这样两件日常事务之间的不同意义。清洁工与老总，这两个角色所关注的对象本身并不同质，因此无法将这些具体事务纳入到同一个体系中去评判价值。

关注位置，是观察这个系统的一个视角。

你在哪里？是谁？在做什么？

作为一个系统的组成部分，如果要观察这个系统，那么就必须清楚三个问题：你在哪里？是谁？在做什么？从这个系统中剔除了“我”，才能分出“其他”，才能分清“我与其他”之于系统间的种种不同影响，从而把这个系统分析透彻。大多数人在做这样的分析的时候，忘记了观察者是“观察-被观察”系统中的一个组成部分，忘掉了“我”的位置，因而少了一半的观察。

《大道至简》这本书通过“工程层状模型”(EHM)，从“实现者”这一角色出发，并论及“团队”和“经营”角色。但是——如同上面的问题一样，EHM模型将这些角色析别出来的时候，也少了一半的观察。举例来说，“实现者”是程序员，“被实现者”是程序，在《大道至简》中却甚少论及程序的本质。又举例来说，“团队”的组成是项目成员，要真正成为“团队”则还有赖于项目目标，在《大道至简》中也只讨论到成员问题，而少掉了对约束或设定目标的那些角色的思考。至于“经营”角色，也存有相似的问题。

反思“工程层状模型”(EHM)的本质，是在讨论一系列工程相关问题在一个轴向上的延伸变化，以及这个轴向的不同领域间的关系。如图0-1所示。

然而这个轴向带来了一个致命的问题：难于承载新的领域角色的加入。换言之，层状模型中加入新的角色就会带来新的分层与界面（关注点），这暗示着该模型下的世界是一对一的、面对面的。这个问题使得我在2006年间陷入了一个困境，不知该如何将“架构师”这个角色放入到这个模型中去。一旦我尝试在EHM模型中为架构