

影响教学服务质量的四种力量

顾学雍 查建中 Edward F. Crawley

【摘要】 本文引用法学教授劳伦斯·莱斯格撰写的《信息空间的代码和法律》一书中所提出的社会行为理论框架,建立一个系统化控制教学服务质量的工作模式。文中定义教育机构为一类提供知识服务的社会机构,旨在向周围的社会环境提供知识积累和传播的服务。莱斯格提出的四个社会约束力模型,在理解技术知识和人类社会之间的复杂关系上尤为适用。在此基础上,我们将教育机构视为一个微型社会,运用四种社会约束的力量对它进行控制和管理,用以实现教育机构的功能目标。我们设计和实施了一系列工程专业课程改革,并以清华大学工业工程系的教学构思过程和运作经验为例说明这一概念。

【关键词】 服务机构 教学服务质量 架构 法律 市场 常态

【收稿日期】 2009年5月

【作者简介】 顾学雍,联合国教科文组织产学合作教席顾问、清华大学工业工程系副教授;查建中,联合国教科文组织产学合作教席主持人、北京交通大学教授;Edward F. Crawley,美国工程院院士、英国皇家工程院院士、瑞典皇家工程院院士、国际CDIO工程教育改革运动发起人和领袖之一,美国麻省理工学院教授、清华大学工业工程系客座教授。

一、概述——莱斯格模型与工程教育的理念

工程教育不仅要提高将提高学生素质作为根本,还要考虑它所面对的社会需求,就是说,它的价值表现在对社会提供知识传递的服务方面。例如一个工程类院校的职责在于,作为一个教育服务机构,它要传播工程技术知识,激发当前和未来的工程师的构思(Conceive)、设计(Design)、实施(Implement)和运营(Operate)方面的能力(CDIO,见<http://www.cdio.org>)。这就要求工程类课程设计的目标不仅要注重学生个人学习技能和理论水平的培养,还要关注整个院系对于社会需求的知识服务质量。所以,良好的课程设计必须考虑到各种社会因素和技术因素。如果把整个教育机构比作一个微型社会,从这个角度去理解,课程设计就相当于为这个微型社会编写一部宪法。目前大多数工程类院校的课程设计主要是由受过工程教育的专家、教授或管理人员来完成,因此,在课程内容的编排上通常是侧重于学生对于技术问题的学习和理解,对于学生应用能力的要求仅仅作为补充,课程内容设计则是面面俱到,追求学科知识的完备性,而很少强调学生如何系统地利用其周围社会环境中的技术和其它资源来解决问题。有鉴于此,我们将法学教授劳伦斯·莱斯格

(Lawrence Lessig)撰写的《信息空间的代码和法律》^[1]中提出的四种力量模型应用于教育机构的社会性服务功能研究,以期有助于提高对教育过程的理解及实践。

1. 何谓四种力量?

莱斯格的四种力量包括:架构、法律、市场和常态。在莱斯格定义中,四种力量成为社会行为的驱动力和约束力。一所学校作为一个教育机构,其行为的影响在于对周边社会提供的知识服务,这也如同CDIO的第一条标准——“情境标准”(CDIO as Context)所描述的。它可以在战略层面上帮助课程设计师看到教育行为的驱动要素和制约因素。下面笔者将结合工程教育环境对于莱斯格的四种力量逐一进行阐述(见图1)。

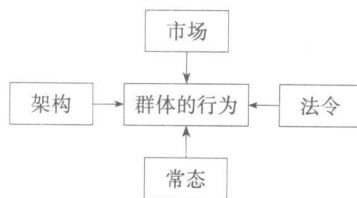


图1 四力示意图(译自劳伦斯·莱斯格原作)

(1) 架构(Architecture)。

架构是贯穿于整个系统的约束,因为架构的建立和影响产生于系统中所有活动发生之前。例

如一个具有密码保护功能的信息系统会具有保护重要信息资源的能力,这就要求用户在使用之前必须取得某些授权,当然也不排除花费大量的智力和时间资源对安全系统进行破解的可能性。在某种意义上这两者是等价的,在使用一个信息系统之前,使用者必须先付出穿透安全系统的代价。总而言之,架构可视为一切行为的前提背景,此背景涵盖一个社会系统的基础建设、人事组织以及工作模式。这恰恰与工程教育设计的思路不谋而合,不仅仅是工程学科的知识结构,整个学院的工作环境(包括各种可用于教学和研究的技术设施及资源),以及学校的组织结构,都可以视为“架构”的一部分。因此“架构”是一种极富战略意义的力量,是稳定或不易变化的系统特征^[3]。相比之下,其他力量如“法律”、“市场”和“常态”等,往往只是因应个别时空环节特殊需要所设计的集体行为约束力。

(2) 法律(Law)。

法律是一套明确的监管规则,用于惩罚那些已经发生的错误行为。例如学校会设立守则或规定用于惩罚那些经常逃课的学生,这种情况下法律效果的有力表达必须满足两个条件:首先,它必须在学生的缺课数量达到应当惩罚的限度时才实施;其次,其他学生必须能够观察到相应的规定是否被正确且得力地执行下去。换言之,法律的力量往往需要不良行为发生之后才能显现出来,而且裁决不公或处罚不当,也不会充分影响到社会的各个层面。法律的权力基于执法行为的一致性。如果执法成本过高,或错判误判太多,法律的价值也不复存在。因此法律运行需要一个具有良好执行能力的机构(如本文所指的工程院校)作为保证。另外法规或政策的制定也要考虑到课程设计的要素。

从时间角度看,架构与法律相辅相成,分别作用在活动发生之前和之后。

(3) 市场(Market)。

市场的力量在于拉动或吸引机构产生某种行为。例如媒体宣传报道学生或教师的成果可以算作是一种市场的激励力量;对于学生来说,被授予荣誉证书或拥有不错的排名和更高的分数,都会促使他们更努力地学习。上述情况即为市场的力量。根据莱希格的理论,市场的力量拉动机构按照一定的方向发展。显然,如果市场的激励没有足够的吸引力,学生或教师也会不屑一顾。总之,

市场仅是一种邀请,它取决于机构本身是否有动力愿意参与其中。

(4) 常态(Norm)。

常态是一种集体的推动力量,它能促使机构的成员参加某些活动。也有人把它称作文化的力量,或同侪压力。比如当所有人在学校中都采用基于网络的版本控制系统来管理共享文件时,这将是一股巨大的力量,可改变那些只使用电子邮件发送文件的个人的习惯。不言而喻,违背常态的人永远会存在,但常态的力量则在于产生积极的推动力,鼓励人们依循常态做事。常态“推动”是远比市场“邀请”更为强大的力量,它试图把个人“非常态”的行为引入正轨,不论个人本身是否愿意这样做。

2. 课程设计的过程纲要。

利用四种力量的框架去进行课程设计,大致遵循以下四个阶段:架构为系统提供一个稳定的框架体系,因此要首先被设计好。一个好的架构设计应具有较长的适用性,否则就要尽早修订,因为所有其他三种力量都依据架构规定的环境来实施。其次是法律,它的确定也会对架构进行补充,因为它需要足够强大的组织结构和技术工具的支持才能实施。市场力量同样重要,它们对人、资源和新的思想产生巨大的吸引力,并整合到统一的教育环境中。常态化的现象则随着时间的推移慢慢积累前三个力量的影响力,一旦常态的趋同力逐渐成型,它将作为一个过滤器,以识别并纠正微观层面那些不符合集体常态的个别行为。

首先,需要认识到不同架构的选择带来的直接后果。例如,当我们要选择一个扁平的组织架构,希望有助于学生自由交流,我们就要考虑到这样一个没有明确团队结构的组织架构,能否有效地支持我们既定的功能和目标。架构选择将对学习的效果产生深远的影响,因此必须在课程开始之前确定。

其次,需要设计和执行相应的监管法律,以保证架构意图的实现。以教学过程为例,这可能包括评分政策、签到政策、课堂参与评价政策和一些应对抄袭等不良学习行为的政策。所有这些政策必须正确、明确、准确,并且得到有效执行,才能提高学生的学习主动性,而不是扼杀学生的创造力。

第三,需要有效的市场激励制度,保证教学本身对学生有足够的吸引力。这就是所谓市场力量。通常情况下,分数和排名可以被视为一种流

通于学校的有效货币。当然还有其他类型的知识货币,如:具有“市场价值”的技能。比如在信息类课程中,一些流行的编程语言和建模工具就被认为是有价值的。因为学生们认为这些技能可以帮助他们用更短的时间解决问题、取得高分或是找到好的实习单位和工作岗位。这类知识资产可以套用市场激励机制进行理解。

第四 所有由人构成的组织都有某种文化层面的常态化规范。利用常态可以在集体中促进有效的互相学习。理想地说,一个积极的学习文化会使学生和教师自动地互相鼓励,提高工作质量,并且变得更富创造性;或者当事情结果不如所期望时大家至少可以在精神上互相鼓励。

二、模型应用——清华大学的案例

为了便于说明,我们以清华大学工业工程系的两门必修课为例,展示我们如何在基于 CDIO 教学模式课程中应用上述四种力量模型。具体地说,就是要探讨如何将一个班级视作一个微型社会,利用四种力量规划和执行整个教学过程和学习活动。

我们选择“数据结构和算法”和“数据库系统”两门课程作为分析案例,这两者本身也具有密切的关系。一方面前者是后者的理论基础,另一方面这两门课作为必修课程被安排在连续的两个学期。这就符合 CDIO 标准 3 关于“集成化课程体系设计”的要求,同时在使用教学工具方面,两门课程都能利用大量的数字化内容和工具参与到教学过程之中。这些具有开源特性的工具可以方便地在互联网中得到(关于这一点我们在下文还会详述)。不同技术的混合使用,使得教师特别容易比较学生在使用不同技术情况下的表现。

1. 架构和功能目标

架构的设计包括三个方面:技术、组织和工作流程。鲍德温等在《设计规则》^[3]中启发我们以这种设计产业系统的社会科学方式对课程规划进行理解。本文提及的技术架构是指课程的技术内容。比如,是以主流产品关系型数据库的基本概念和应用作为主线,还是在课程开始的时候详细讲解编程语言及一般数据库的基本概念以建立完整的数据库概念和理论?或者这可以抽象为“做中学”还是“听中学”两种模式。这两种完全不同的方法决定课程的不同技术架构,并且这种不同可能从根本上影响学生的学习方式。组织架构指的是学生和教师所构成的教学组织及其在组织中

的关系。例如,学生是教学组织的中心,是主体,教师是指导,是教学的组织者和管理者;学生可以组成小组学习和工作;而传统的教学模式下,教师是主体,学生是被动接受知识的客体,学生多是单兵作战。显而易见,两者具有本质性的区别。工作流架构指定了不同活动、物理设施和信息流之间的互动模式。具体到每周的课堂学习、每两周的小组会议和小组项目的终期报告等一系列学习活动的准备和开展,都是典型的工作流模式。这些简单的条件实际上涉及到人和物之间复杂的相互作用。

技术架构、组织架构和工作流架构都必须符合一套能够统筹全局的功能目标。在决定采用何种架构之前,我们的首要任务是邀请学生与教师设计适用于小组学习方式的学习制度。整个设计的过程也是一种挑战,我们提出了整体的教学目标:

A. 积累知识。

a. 积累人力和资源,以便更好地保留现有的知识。

b. 总结经验和传统,作为一个技术孵化地。

c. 汇集对同类知识的兴趣及后续发展的动力。

B. 传播知识。

a. 向学生传播知识。

b. 在不同院系和学校之间传播知识。

c. 在社会的相关机构传播知识。

C. 创造知识。

a. 增量式创造。

b. 跨学科创造。

c. 革新性创造。

D. 运用知识。

a. 在学习过程中应用知识。

b. 经过分析、批判及整合的知识消化过程后应用知识。

c. 团队合作、综合应用跨学科知识。

当然,上述目标并不是硬性指标,比如传统上管理人员使用量化的方式度量各个指标是否达到,又如许多学生在校的目标是考试取得高分和获得学位等等。这都不是我们希望看到的。为了向学生和教师表明学习目的是获取和运用知识,而不是得到一些抽象的数据,我们决定从应用知识的角度表述这些目标,并把它们作为指导方针,以帮助它们设计自己的学习活动。

根据上述目标,不难发现,我们的宗旨在于建立一个通用的学习型组织,这里称之为“知识城”,简称为“KC”。起初,“知识城”只是我们数据库课程中的一个小组项目,用以整合其他小组的团队项目。后来我们发现这个目标具有普遍意义,每个院系的教学目标亦应如此。这些目标也就成为课程指导的重要组成部分,因为它们提供了一个思路,以便组织以小组为核心的互动学习活动。它们又可以充当灵感来源,帮助学生解释为什么他们选择的项目在课程中具有价值。所有那些有趣的项目都需要特定领域的知识和来自小组外部的支持。这些支持有的时候甚至来自校园之外。

这些目标也有助于在课程设计的同时完成架构设计。比如对一个课程的知识内容进行讲解时,最重要的事情在于如何明晰地表现主题。对于学生小组的激励不是告诉他们击败其他小组,而是围绕统一的目标进行合作。因而在设计学习流程时,要围绕这一目标设计各种互动,对各个小组进行检查和督促。所以这些目标在课程一开始就帮助所有的参与者选择适当的技术架构、组织架构和工作流架构。

2. 架构设计。

每一个学习活动都必须为我们的课程目标做出贡献,体现在知识积累、传播、创造和运用方面。这些活动也受限于系统预先确定的架构。在本文中,我们重点关注下面三个概念,即技术架构、组织架构和工作流架构。下面的章节可以解释如何在我们设计的两门课程中发挥架构的作用。

(1) 技术架构。

数据结构和数据库课程的主要目标在于积累、传播、创造和运用计算机设备操作、存储和表现数据的知识。在著名教科书《计算机程序的结构和解释器》^[4]中,我们发现关于数据结构和计算机程序的描述可以建立在一个相当简单的准则之上:如何使用计算机将数学上的抽象概念近似地表达具象的事物。为了确保学生可以快速理解这些抽象的数学概念,书中使用 Lisp——一种解释性编程语言,将学生置于具象的计算世界之中。这本教科书提供了一个类似课程的技术结构。从教学的角度来看我们可以得出这样的结论:两门课程的学习活动的设计,应围绕如何使用计算机技术来说明和表现抽象的数学概念与具象的应用之间的联系。因此课堂讲座按照抽象的数学概念进行组织,具体到课程中就是不同的抽象数据类

型(Abstract Data Type, ADT),如集合、列表、树和图等,以此表述其内在联系。然后 ADT 成为在课堂上描述概念和例子的“官方词汇”。而知识积累、传播、创造和运用的技术架构就建立在描述知识的科学语言之上。学生在课堂上掌握更多的词汇能够扩展自身的能力,表达自己的技术思路。比如当他们了解了“排序”和“搜索”之后,不仅扩充了自己的技术语言,还能进一步了解相关的度量指标,如计算时间复杂性和内存空间的消耗。当他们不断了解不同的编程语言和数据库技术时,词汇量也不断增加,使他们能够从不断理解抽象的数学概念进入到更加丰富具体的技术应用。我们总结成一句话:科学语言的词汇、语法及其应用背景,决定了工程知识的技术架构。

课程中涉及数据结构、算法和数据库的部分一定会有编程的任务。对于第一次接触数据结构课程的学生,这也可能是他们第一次接触到现实世界软件工程的概念。很多学生在编程时考虑的仅仅是任务要求,而不是从软件工程项目出发。针对这一情况,我们以编程的任务为契机开展软件工程实践。为此我们在校园网内部署了基于 SVN 的版本控制服务(<http://subversion.tigris.org>)。它将不同作者的文件、源代码进行同步处理,与其他数字信息资源构成一个中央资料库。同时,比较和记录了每个文件不同版本之间的变化,由此形成一个基于互联网且模拟现实工程环境的软件开发社区。它整合了各种不同的开源工具,并且自动分析不同版本源代码之间的变化,包括作者、变化频率、变化数量和变化的正确性。因此,使用版本控制技术所提供的技术平台成为分数评定的重要组成之一,并且能够检测出剽窃、迟交、赶工的行为。根据上述目标,我们使用 SVN 实现了学生学习进度管理的透明化。试想,如果没有这些开源技术和现有的网络基础设施的保证,这一效果必然需要大量频繁的师生互动才能够实现。

我们使用开源课程管理的网站系统 Moodle (见 <http://moodle.org>),为学生提供博客、日历和文件共享服务。教师和助教可以在网站上对学生布置家庭作业、发布讲义等。我们选择 Moodle 主要因为它是一个成熟的开源产品,并拥有一套完整的社会网络模块用以满足各种复杂的沟通需求。作为技术基础设施, Moodle 也有利于我们以一种更细致的方式观察学生的学习活动。比如

我们可以看到谁是最热门的博客、谁经常与他人分享信息等等。对于那些不积极的学生,就能够有针对性地安排面谈,了解他们的具体问题,以推动他们的工作进展。

更高层次的“技术扫盲”来自学生在实践中的直接经验与新的技术基础设施的引入,并且通过师生及其与技术之间的互动得以解决:版本控制软件为他们自动生成项目版本号和分支名称,管理代码的进化历史;Moodle让学生能够在线组织团队讨论,使用多人协同的系统发布问题,并在不同的小组之间分享知识。这些技术扩展了学生日常交流的范围和方式,而其结果不仅仅使他们说话像(软件)专业人员,也提高了沟通具体想法的有效性和效率。

(2) 组织架构。

组织架构是一种制约的力量,用以规范人们相互交流的方式和范围。根据上文的技术架构部分,学生把自己看成是“知识城”的公民,使用共同语言进行交流,并借助技术工具和架构创造了完整的社会体系。在“知识城”,教学人员是每个班的政治家或市长,他将任务布置给副市长也就是助教。助教负责观察学生小组和学生个人的学习习惯,他们不仅要解答技术方面的问题,还要使用上面提到的技术工具来分析学生的个人行为。他们观察在Moodle上个人和小组发帖的频率和问答交流(FAQ)的使用情况;他们还要检查学生是否在中央代码库上进行足够充分的上传和修改自己的源代码。学生则组织成小组,每个小组扮演某个服务机构或产品制造商。在学期初,每个小组会设计自己的名称和徽标,最初两个星期必须提出如何在“知识城”赢得潜在客户的方案。多数情况下,他们的客户是同一个班的其他小组。然后每个小组将根据自己的情况自行设计和实施组织架构。

在基于项目的学习中,要求整个小组共同完成产品的开发。以这两门课程为例,这些产品包括:小组形式的学生演讲、小组项目进度报告、个人课程作业、随堂测验和小组项目。根据我们的了解,许多学生没有足够的相关知识和经验,因此需要周期性地整合小组中每个人的工作。通常情况下,学生小组只是列出一个任务表进行分工,然后分头实现直到任务完成,这看起来可以很容易地组装成一个具有完整功能的产品,也不需要赶工或是加班。基于这种心态,学生通常各自为政,

不幸的是,往往在组装整合的阶段才发现存在许多严重问题。

为了避免这种幼稚的项目开发习惯,在组织结构上就要让小组认识所处的环境,所有的产品和技术内容交付必须能够满足系统整合时的各类需求。学生必须反复与协作的同伴以及教师和助教进行沟通,从而得以完善其团队项目。在“知识城”中,每个小组必须明确自己的技术优势和能够对城市提供的技术贡献,因此有责任在所有的口头和书面报告中解释项目对于“知识城”的社会价值。我们同样鼓励小组之间的技术整合。“知识城”的概念有助于学生理清思路,以关联小组之间的各种商业和社会互动。项目的意义并不在于塑造出一个最出色的小组,而是建立由许多互相依存的产品开发小组构成的和谐的社会体系——“知识城”。

两门课程各有60到80人参与,通常组成10到12个小组,每个小组5到9名学生。同一系内的学生大多住在同一宿舍,并学习同样的一套课程,这就使得我们比较容易协调小组的时间表。有时会有来自其他院系的学生,但由于属于少数群体,因此只能服从对多数人的调度安排。考虑到两门课程的工作量较大,教师也要在院系内部派任,这样他们就能切实了解学生的总体工作量,而不会与其他课程的教学任务发生冲突。不言而喻,上述安排的目的在于让学生愿意且拥有精力投入到“知识城”的项目当中。

(3) 工作流架构。

技术架构决定了要学什么;组织架构决定谁来负责哪种类型的学习活动;工作流架构决定如何去进行学习。利用“知识城”的背景,学习行为主要发生在课堂讲座之前和之后。整个工作流程如下所示:

- A. 课堂教学流程(每星期3课时)。
 - a. 教师讲座。
 - b. 学生进行上周的回顾与审查(3分钟录像)。
 - c. 学生演讲(利用小组项目作为应用背景讲解技术内容)。
 - d. 随机挑选小组作进度报告。
 - e. 课堂测验。
- B. 内容准备工作(以周为单位循环)。
 - a. 确定讲座主题(根据教学大纲,上周的小组参与组内讨论)。

- b. 制作 alpha 版本讲义(故事)。
- c. 与教师和助教进行第一次彩排。
- d. 制作 beta 版本讲义(详细内容)。
- e. 与教师和助教进行第二次彩排。
- f. 在讲义的基础上编写随堂测验内容。
- g. 上传所有的内容和资料到源代码库中。
- C. 内容精化工作流程。
 - a. 课后马上与教师对课堂进行回顾。
- b. 创建勘误表清单。
 - c. 制作回顾讲义,包括:上次演讲的重点难点,上次演讲遗漏和错误的知识点,其他参考资料,3 分钟的视频表现课堂上的精彩瞬间。
- d. 组内排练检查。
- e. 上传所有的资料到源代码库。
- D. 每周小组项目进度报告工作流程。
 - a. 根据每周的讲座调整小组项目的发展方向。
 - b. 继续发展小组项目。
 - c. 制作进度报告和相关技术展示(5 到 9 小时幻灯片)。
 - d. 上传进度报告和技术演示到源代码库。

值得一提的是,上述工作流程也是由学生整理出来的,它不能涵盖所有的细节,仅提供指导性意见,用以在不同的时间尺度上规范和制约学习活动的开展。工作流规范了那些与架构准则相协调的学习活动。所有“知识城”的课程相当于运行着的多道程序。当一个小组建立了一个展示,其他小组可以利用在此之上发展的技术将其他小组的成果作为自己项目的引例。如果本周的演示文稿没有来得及反映这一变化,我们也鼓励在下周上课时反映在成果报告或回顾内容当中。我们还鼓励学生引入其他课程的内容。因此每周课堂演讲是所有成员的同步点。类似于城市的市政厅会议,他们在其中不断地突出各种学习成果和新的机会,而这些机会往往要在会后通过不断的碰撞与检验,才能变成新的成果。

为了使工作量能够平均分配,每个小组至少有一周要负责技术知识学习的任务,当周将学习的结果在课堂上对全班进行汇报。对于不能达到最低标准的小组,要求再做一次。重要的是当周的小组表现将会作为下周的最低标准。这样,通过累计实现质量改进,不仅促使小组必须与之前完成的小组进行经验交流,也自然会促使学生课后进行高质量的准备,而这种难度不断提高的标

准又迫使后面的小组提前开始准备。根据学生的反馈,有些学生甚至提前了整整一年时间就开始了与课程相关的准备。工作流架构强调逐步改善和反复进行,其同步时间点帮助学生更好地管理自己的时间。工作流仍然强调考试成绩和排名,但目的是将测试和分数作为诊断和持续改进学习效果的工具,考试不再是“应试”模式下纯粹的学习目标。

3. 法律设计。

架构建立活动前的限制,法律规定活动后的约束。举例来说,基于互联网的源代码控制系统提供了一系列技术手段,监测作业提交的作者和时间。这也要求每位学生必须获取用户名和密码,才可以参加网络社区。然而技术架构本身无法提供任何机制确保学生准时提交作业。在课程中使用评分政策,利用法律的效力直接影响学生执行,会快速收到成效。

执法的过程会耗时费力。理想情况下,一个良好的架构不仅会减少违法行为的发生,而且还有助于法律的执行。例如,版本控制软件能够自动检测迟交行为,也可以通过执行源代码对比发现剽窃行为。法律或分级政策应该在架构决定后再去建立,并在良好的技术、组织和工作流程架构下执行。

法律的力量应当源于自发性推理的能力。人们在违反法律之前往往会意识到自己的行为在触犯法律之后将承担的后果,这种推理能力驱使人们避免实施可能犯罪的行为。还有一种情况,就是无意识的犯罪,这时推理的能力将失去作用。比如抄袭,许多学生往往是不知道如何使用正确的方式引用外界的信息,这种无辜且无知的行为使得他们经常在没有察觉的情况下触犯了“盗用智慧财产”之法。为了在“知识城”中解决这个问题,我们在知识的积累、传播和创造过程中,通过不同种类的活动方式,完善学生的知识结构(技术架构),让学生们逐渐认识合法与非法之别,避免无意识犯罪。

在知识积累过程中,学生通过对该领域历史知识的认知去完善自我知识结构。在数据结构和数据库这两门课程中,学生需要知道一些重要算法的发明者、什么工具用得上这些算法以及为什么要使用这些算法。学生对于过往知识的理解有助于合理合法地进行创新,规避对前人专利和知识产权的侵犯。

在知识传播过程中,传播的有效性可依据传播内容的可用性来进行评价。首先,需要将技术内容组织成一定的格式,从而使接受者容易理解,并可将来知识按照接收的方式传播下去。不合理的组织方式通常会影响阅读人的理解,从而损害信息传播的完整性。还有就是学生必须知道当前内容能否合法地进行传播。例如,在数据库课中,我们会组织讨论什么样的数据是不安全的,为什么?什么样的隐私信息是我们不应该共享的?版权法限制我们不能传播哪些信息内容?通过这个过程,提高了学生知识产权和版权的意识,减少了无意识的侵权行为。

鉴于拉托尔的科学发 展论^[3],在知识创造的过程中,我们要求学生 对创造的知识进行群体化论证,以确认创造的合法性。这项工作集中发生在学生讨论小组项目选题的时候。学生阐述小组的项目选题与专利申请或法庭上的某个程序非常相似。验证一个创意的有效性和合法性需要涉及许多“证人”。它需要学生在知识积累过程中完善个人的知识结构。这些知识必须以严密的方式进行组织和证实,否则知识创造就会失去控制。如前所述,法律的力量来自于推理的力量。学生需要学会保护自己,避免不必要的违法行为。当然整个过程也需要教师的参与,事实上在教师办公室经常发生类似的师生讨论,因为这些创造性的想法需要时间来挖掘和论证。上述学习活动帮助学生拓展了在工程学科的诉讼技能,认识到如何申明自己的知识贡献,系统性地减少了无意识侵犯“知识城”法律的行为。而对于一些蓄意违法行为,如作弊或怠工行为,在法律的推理力量之外,我们还需要市场激励力量和常态推动力量进行配合矫正。

4. 市场设计。

在大学环境中,学生往往想获得高分,以提高他们的平均学分成绩。教师或许会关心学生的教学反馈,因为这些反馈经常与教师升级或聘用条件挂钩。助教通常忙于科研项目,因为研究生需要新的想法并物色志愿者来帮助他们进行实验。我们认为,在“知识城”中,分数、学生的教学评估意见和时间开销属于不同的货币。在不同情况下,这些货币彼此之间有不同的汇率。本节说明在两门课程中货币的度量和使用。

(1) 评分政策。

对于大多数清华大学的学生而言,分数至关

重要。他们需要优异成绩申请免试直推研究生。排在前百分之五十的学生能够享有学校的各种奖励。分数是一个非常具体的货币。它直接规范学生的行为以满足教师的评分标准。从这个意义上讲,教师和助教拥有分配货币的权利。通常情况下由教师在课程大纲中明确每一个作业或是任务在课程分数中的权重,学生不能参与大纲的制定。

“知识城”的教学大纲更像是教师和商业合同。我们召集参与过和将要参与课程的学生修改教学大纲,并在每个学年的开始征集学生的反馈。我们希望教师和学生考虑大纲的评分标准是否可以被接受。而这种学生参与的方式传递了一个重要的信息:评分标准只是一种机制,学生有机会决定自己的分数和评分的机制。这一过程能促使他们不断地理解学习的目标——追求的不是分数本身,而是能否在一系列不同的学习活动中获得更多的价值。我们指导学生拿出评分标准用以实现“知识城”的目标。他们很快就意识到标准与目标之间的关系。分数是一种非保守性资源,不同于石油或黄金,分数可以基于人为的意识而自由分配。如果多数学生都非常努力,那就都有机会得到及格的分数。为了区分他们的成功程度,可以让那些贡献较大者获得较高分数。这样的评分制度及其建立过程激励学生更加注重学习成果,学习的目标是获取知识,而不是为了高分而博弈。

(2) 学生反馈。

在许多学校,学生会在学期结束后对教师的教学进行评议。由于学生的教学反馈会直接影响到教师的职位晋升。有的时候为了取悦学生以获得高的评价,教师可能降低学术方面的标准,由此产生一群高分低能的学生。这样的师生互评方式不一定能激励教师进步,反而有可能削弱教师的权威。

为了保证和提高教学质量,清华大学工业工程系在系主任萨文迪教授和常务副系主任郑力教授带领下,于2008年春季学期在整个院系开展了创新性教学的实验计划。他们选择三门课程进行实验,测试不同的讲课方式对学生创造力的影响。在实验期间,校方教务部门同意不使用学生的反馈意见作为教师晋升的评判依据。这一特殊政策在数据结构课程试用,而且在这个学期对所有参加的教师和助教增加了额外的财政奖励以补偿实验所需的额外工作。这一政策显示了行政体系支

持教学创新的意愿,也证明各类政策是可以根据教学需求而改动的。

在“知识城”,我们同样认为学生的反馈是帮助我们改进教学实践的重要因素。但是我们不希望把学生和教师置于互相牵制的境地。如果奖励制度设计不当,往往会扭曲了原来的教学意图。所以,只要有合理的理由,教务体制可以进行调整以利于支持学生创新。我们相信在“知识城”尝试的种种概念能够有助于设计更好的师生互评方式,包括学生对教师的反馈和教师的奖励制度。例如在“知识城”中学生应邀来帮助设计评分标准。同样,我们也希望进行更多的实验去发现更好的货币交换策略。

(3) 时间承诺。

学习是一个非常耗时的过程,设计一个新的学习系统更是一个耗时的过程。它需要很多的时间思考并完善整个系统来提高学习的有效性。时间作为可以市场化的货币,提高时间的可利用价值就会产生强大的市场吸引力。最初建立“知识城”时,不论是过程、服务和基础设施都必须根据当时的环境和可用的科技资源重新检验和设计。这要求学生、助教和教师利用他们的个人时间从零开始建立项目。大多数工程项目最初的尝试可能会遇到挫折,而且整个过程没有任何物质上的诱因,可以说是一个纯粹以“学习”为主题展开的创新行为。我们常常开玩笑说,大学的行政结构是人类社会中变化最慢的部分。而工程院系不应该把资源花费在高度抽象的概念,如“如何帮助学生认知抽象理念”等形而上的哲学探究活动。工学教授和学生应把重点放在具体的应用学科,因此在时间上会同时表现出技术性和务实性的价值。

在不同的市场,价值判断是不同的。工业工程的价值是在全系统整体规模上找出改进的机会。当你想建造一枚火箭时,你需要专门的火箭工程师;而当你想建造 1000 枚火箭时,你就需要工业工程的原则。同样的道理,教一个学生,需要找到一个好教师;教 100 个学生,就需要教师们运用工业工程的原理来优化教学组织和过程了。在清华的工业工程系(Department of Industrial Engineering)和麻省理工学院的工程系统学部(Engineering Systems Division),我们的研究兴趣就在于人与技术之间复杂的相互作用。也就是说,设计学习组织的工作在研究系统工程的工程院系中很容易展开。而我们认为,所有其他工程院系也应

关心人和技术构成的复杂系统之间的相互作用。它可能给各部门带来一个新的视角,丰富知识市场中的解决方案。

在“知识城”中,所有的学生和助教都被明确告知他们的目标在于提高整个学习型社会的水平。各参与者由于市场的吸引力,愿意将时间用于“知识城”的建设,并利用数学方法来衡量各自贡献的效果。

我们要求学生参加以小组为单位的专题报告会和开展小组研究项目,使每个人都能分享他人的经验,学习新的思路和新的工具。当共享在技术工具支持下成为一种普遍做法时,学生也开始受益于自己的社会贡献。那些贡献较多的学生得到了其他人更多的尊重,其成果都保存在数据库中,还能有机会得到高年级学生的帮助。最有能力的学生可以协助教学工作。教师可以集中精力在课堂上讲授更精深的想法。总之,在“知识城”市场上,“知识成为主要的货币”,有付出就有回报。这种良性循环不断刺激每个参与者投入更多的时间,并获得更高的市场价值。

5. 常态的设计。

社会的趋同性推动所有成员养成一定的习惯。正是这种常态化的趋同性产生了社会约束力,而且可以利用它来提升学习效果。在“知识城”,趋同性使得大家如同时装表演一样,在统一的形式和舞台下不断创新,带来惊喜。在城市会议也就是课堂讲座中,每个小组拥有机会来展示他们最新的“时尚”产品。每个学生的演讲必须在独特性和创新性上不输于前一小组。否则成员个体会感到组内的压力,部分原因是因为他们的表现会反映在小组的分数中。如果一个小组或个人的表现很糟糕,他们在整个群体中会感到很尴尬。这与社会常态中的趋同性有关,或者说,他们会面临得不到某种文化认同的压力。

(1) 与众不同的权利。

我们对那些通过灌输的方式将学生同化为一种类型的人的训练手段不感兴趣。工程师的能力在于能够将跨领域的知识和技术整合成问题的解决方案。理想的情况下,我们希望学生采用同侪压力的方式,自我驱动地为团队做出自己独特的贡献。也就是说,我们试图创造一种文化常态,这种文化提倡突破自我,不断创新。我们不愿建立一种习惯,让学生盲目地认为某些工具或方法比其它的更好,而是允许甚至鼓励学生尝试不同的

工具和方法,譬如混合地使用 Java、Mathematica、C 和 C++ 完成小组项目。我们建议他们学习和使用某种编程语言如 Java,部分是考虑到它易于推广的特性和流行的程度,可以让这些刚刚接触代码的学生利用各种教程、文件和参考源代码来协助学习,但这并不意味着他们只用 Java 解决所有问题。当学生使用 Java 或 C++ 时,他们会把问题的解决归为软件工程的问题。他们需要设计类图并寻找合适的源代码库。这迫使他们花费大量时间学习语法和相关的类库,而一旦他们在源代码上花费太多时间,就会忘记工程的目的是为了解决应用问题,而不是展示编程技巧。特定软件工具能以其规范和使用者的习惯有效地解决某些问题,但不会适合解决所有的问题。

为了让学生从 Java 以外的视角理解计算科学,我们还引入 Mathematica 作为编程和演示的环境。作为一种迅速构建计算实验的平台,Mathematica 提供了以互动方式解决问题的思路,而作为一种解释性语言,它提供了庞大的类库和可视化编程组件,对于很多需要快速实现原型的任务,它能减少许多阶段的软件工程开发的工作量。我们希望学生学习这种不同风格的编程技巧,使他们能够了解处理这类工程问题的常态做法。当应用场合适当时,这些常态的规范做法是 very effective 的。由多个视角与工具所建构的视野,将使学生把创新视为生活常态,从而形成经常性的内在压力,促进个人快速成长。

(2) 共享原则。

软件自由化基金会创始人理查德·斯托尔曼^[9]提出如下观点:“免费软件是一个自由的事物,只要是有益于社会的方式,所有人都能够去使用。”这一理念引发了软件工程革命,在此基础上开发了 GNU,俗称 Linux,世界上第一个“开源”的操作系统。它建立了全新的社会常态,使大量的工程技术可免费使用。

我们认为,这种价值观必须深深植根于“知识城”。在很多热门的 MBA 课程中,“拓展竞争力”的常态化社会价值远远不同于“有福共享”的开源社会价值观。我们想要建立这样一种工程师的文化价值观,那就是:如果需要创造更大价值,就要在更大的社会群体中共享各类工程方法和技术资源,并且接纳统一的开源标准。从操作层面看,世界已经受益于软件自由化基金会所营造的哲学和社会常态。许多技术免费提供给公众,创造了众

多平等的技术社区,最重要的是这个简单的哲学产生了许多高品质、低成本的产品。从教育资源的层面看,如果没有理查德·斯托尔曼的技术成就和他倡导的社会常态,我们所用的课程工具会大大减少,用开源科技支持的学习活动亦将难以实现。

三、结论——从常态到架构的回顾

现在从常态到架构进行反向回顾。在 21 世纪,我们见证了这种共享开源的文化在不同的领域和市场中都创造了大量新的产品。而这种对于开源产品和服务的认可也改变了相应的法律,从而迫使法律从业人员使用新的方式来解释所有权的问题。工程产品成本的降低也改变了架构的实际,以实现知识积累、传播和创造。我们认为这种以社区为基础的学习活动由这些力量所驱动,课程的设计不应只着眼于具体学科的技术内容,他们应该将一个班级或院系的学生作为一个微型社会来看待,进而利用该社会提供的所有机会与潜在力量,创造一个学习型的互动环境,从而形成累积、传播、创造与运用知识的功能系统。

1. 四种力量与课程设计。

基于莱斯格的四种力量模型,本文提出了设计系统导向的工程课程的一系列战略构想。我们认为,课程规划必须从设计架构开始,课程设计师要去设计技术的、组织的和工作流的架构。一旦架构决定,我们就可以利用这些架构的特性来执行法律。只有通过有效地执行法律或教育政策,法律的力量才能够加强整个社会的学习效果。否则法律只是口号和文字,没有任何制约力。

架构和法律是在活动发生前后的两种相互调和的力量。市场和常态是推拉整个学习型社区的两种相互聚合的力量。这四种力量互相补充,提供了一个新的课程规划和执行框架。过去工学教授与资深专家往往以一种个人学习的方式设计课程,在我们的课程体系中,学生构成一个微型社会,其现实背景是学生必须在毕业之前实践各种合作、谈判和其他的生存技能。为了确保有这种微型社会能够有效地给学生提供有益的工程经验,我们描述了四种力量模型在“知识城”中使用的形态和方式。虽然我们的经验只限于一所大学的一个工程系,但我们认为,本文的许多内容可以与其他院系和大学共享,并有可能减少他们的学习代价。

2. 对称与平衡。

首先必须承认,上文提到的两门课程不能算是一个完整的课程体系。然而我们认为,本文提到的“对称”原则和方法将适用于不同类型和规模的组织。换言之,即使应用到更为复杂的环境当中,这些力的运作原理也不会改变。值得一提的是,莱斯格四个力量模型的最初设想来源于使用互联网的全球化社会。因此我们对这些力量可以用来规范许多不同规模和类型的人类活动充满信心,其适用性包括宏观、中观和微观的社会组织。

行政组织结构的目标往往集中于检查和制衡。当我们想动员整个社会提供互相学习机会时,我们需要涵盖所有发展可能赖以形成的对称。架构与法律是一组在时间轴上相辅相成的力量。市场和常态是一种镜面相对的制衡,由市场来拉动,以常态去推进。这些力量驱动学习社会的质量不断提高。四力的模型提供了一个简洁的概念框架,描述了社会中的各类复杂现象,使预期的目标和设想能有序地贯彻执行。借助社会学家的理论,我们或许能使工程技术知识的累积、传递、创造与运用过程超越纯艺术的层面。

(感谢:本文的主要内容曾于2009年6月在新加坡召开的CDIO全球年会中以英文发表。在经过原作者群修正与补充后,现以中文发表。感谢原苹果电脑亚洲区教育战略部门负责人阿兰·格林伯格(Alan Greenberg)先生,他不断地为本文

注入鲜活的想法。感谢麻省理工学院工程领袖计划教育单位负责人戴安·萨德霍姆(Diane Sorderholm)博士,她帮助我们正确理解那些关键的技术概念。王天居同学将英文的原稿翻译为中文初稿,何永汕女士为中文稿进行了文字加工。还要感谢Wolfram公司,为“知识城”所有学生和教师提供了以Mathematica为主的软件工具。本文的部分研究工作得到国家自然科学基金资助项目70771059的赞助。)

参 考 文 献

[1] Lessig, L., Code and Other Laws of Cyberspace, Version 2.0, Basic Books, 2006.
[2] Koo H. Y., “A Meta-Language for Systems Architecting”, Doctoral Thesis at Massachusetts Institute of Technology, 2005
[3] Baldwin C. Y. and Clark K. B., Design Rules, Vol 1: The Power of Modularity, MIT Press, 2000.
[4] Abelson H., Sussman G. J., and Sussman J., Structure and Interpretation of Computer Programs, 2nd Edition, MIT Press, 1996
[5] Latour B., Science in Action: How to Follow Scientists and Engineers through Society, Harvard University Press, 1988.
[6] Free Software Foundation, “About the Free Software Foundation”, <http://www.fsf.org/about/> web page, last accessed on May, 24, 2009.

Four Forces Influencing the Teaching Service Quality

Gu Xueyong, Zha Jianzhong, Edward F. Crawley

In this article, a working mode that systematically controls the teaching service quality is established by quoting the theoretical framework of social behavior proposed in the book Code of Cyberspace and Law written by professor Lessig Lawrence. Educational institutions in the paper is defined as agencies in a service industry that provide knowledge accumulation and transmission services to its surrounding society. It can be found that Lessig's four-force model is particularly suitable for understanding the complex interactions between technical knowledge and human societies. Under the circumstances, educational institutions are treated as miniature societies, and the four-force model is applied to control and manage them so as to realize the function targets of educational institutions. Besides, a series of reforms of engineering courses are designed and implemented, and they are illustrated by showing the teaching process and operation experiences of the Department of Industrial Engineering at Tsinghua University.