

面向计算机专业的软件开发技术课程集成发现教学法探索

王小平 裴喜龙 卫志华

(同济大学电子与信息工程学院计算机科学与技术系 上海 200092)

摘要 在调研和分析国内外高等院校计算机和软件工程相关专业课程体系及其存在问题的基础上,面向信息技术高端人才需求和新旧体系调整的要求,基于软件开发中的抽象思维(对象思维、模式思维和架构思维),提出了以运用能力为主要目标的集成发现教学法和教学模式。并且,进一步探讨建立以软件开发方法及其综合实践为核心的应用软件课程教学体系,重组基础课程、核心课程和相关平台应用课程关系,使这些课程内容联系更加紧密,衔接更加顺畅,有利于学生运用能力和创新能力的培养。

关键词 软件开发技术 课程改革 集成发现教学法 抽象思维

中图分类号 TP3

文献标识码 A

DOI:10.3969/j.issn.1000-386x.2017.09.005

RESEARCH ON THE INTEGRATED DISCOVERY TEACHING METHOD OF SOFTWARE DEVELOPMENT TECHNOLOGY CURRICULUM FOR COMPUTER SCIENCE MAJOR

Wang Xiaoping Pei Xilong Wei Zhihua

(Department of Computer Science and Technology, College of Electronic and Information Engineering, Tongji University, Shanghai 200092, China)

Abstract There are some problems with the computer software engineering curriculum system in our investigation and analysis of domestic and foreign universities. In order to meet the requirements of high-end talents for information technology and the adjustment of old and new systems, based on the abstract thinking (i. e. object of thinking, design pattern thinking and architecture thinking) in software development, we propose an integrated discovery teaching method and teaching model aiming at main target of the ability to use for undergraduate students of computer science. Moreover, to further explore the teaching system of application software course based on software development methods and comprehensive practice as the core, the relationship between application of reorganization of courses basic courses, core courses and related platform, make the curriculum more closely, converge more smoothly, which is conducive to the cultivation of students' innovation ability.

Keywords Software development technology Curriculum reform Integrated discovery teaching method Abstract thinking

0 引言

随着信息技术和计算机学科的发展,软件系统的重要性在整个计算机系统中所占的份额及其本身的复杂性迅速地提高,使得它的开发代价和重要性超过了基本硬件系统。目前,相对于信息技术发达的国家而

言,我国信息技术及其产业的发展面临“技术少标准、开发缺规范、生产无检验、质量无保证”落后压力。本科生教育是信息技术人才的主要源头,问题求解、程序设计、软件开发和创新能力是计算机专业人才优劣衡量标准中最重要的方面之一。然而,软硬件能力的培养并非一两门课可以完成的,而是需要一个长期的、有目的的修养过程。系统及其运行信号的非直观可视

性,又使得这种能力在很大程度上依赖于逻辑思维方法、抽象思维能力,以及对学科发展过程中不断深化的自顶向下、逐步求精、分而治之等思想,依赖对计算理论、抽象建模、系统设计和实现三个过程的基础内容掌握程度。另外,不断涌现的新问题求解、新系统设计和创造、新技术应用,也凸显了本学科创新思维和能力培养的重要性,要求学生目前的软件开发技术和研究发展状况有很好的理解和掌握,以适应当前应用及未来发展的需要。计算机专业教学内容和形式也在发生深刻的转变,要求传统的“教导”或“灌输”、自底而上的单一展现模式,逐渐构建为自顶而下的“集成式”、目标导向的“发现式”、问题求解的启发式,甚至是这些模式的混合体。这些教学方法和教学模式的研究、发展和逐渐成熟,将会受到广泛关注,作为教学方法论的参考,也会逐渐被开放式背景下类如 MOOCs 和 SPOCs 在线形式教学方式所接受,使得更为广泛的受教育者获益。

2013 年底 IEEE & ACM 联合发布 CS2013^[1],这一版本与 CC2001 和 CS2008 比较起来,在知识体系有了较大的变化。CS2013 包括 18 个知识领域,其中软件开发基础(SDF)和系统基础(SF)作为两个新的知识领域被正式提出,并列出了各种不同类型大学的 80 个课程样本和 5 个完整的课目样本。现有的课程包含不同知识领域的论题,可以用不同方法组合成一个完整的课目,在课时数满足的前提下覆盖所有课目的知识单元。

计算机科学与技术作为一门学科的边界很广阔,学生兴趣又各有不同,这就要求专业的课程体系为学生在本科阶段的学习提供不同的选择。但无论如何,学校课程的教学体系也有责任保证,不论学生的选择是什么,都能学到计算机系统的核心知识和主要专业技能。例如,软件开发方法学作为一门新的专业课程在国内许多高校相继开设,包括天津大学计算机专业自 2011 年开始开设“软件开发方法论”,北京大学、西安交通大学、西安电子科技大学等都先后开设了类似的课程,相应的教材也开始编写。但系统性不强、内容陈旧,应该与非计算机专业的《软件开发技术基础》教材要求有较大的区别。不仅要求学生掌握软件开发各个阶段的技术方法,还要求掌握软件开发全过程的技术方法,这是计算机专业培养系统能力的重要入口,如软件建模、构件技术、软件架构和基于平台的开发技术,并且强调可尽量采用新的技术、管理方法和新工具,与软件开发技术的行业标准相适应。北大青鸟软件开发培训课程作为国内 IT 职业教育的知名品牌,迄今已持续十多年,一方面说明不同层次软件开发人才

需求旺盛,另一方面也说明了目前学校专业教育的短板。只有在理论、技术和实践能力都有较高水平的教学要求,并付诸实施,才能满足目前信息产业高端人才的紧缺需求。

本文在调研和分析国内外高等院校计算机和软件工程相关专业课程体系及其存在问题的基础上,面向信息技术高端人才需求和新旧体系调整的要求,基于软件开发中的抽象思维(对象思维、模式思维和架构思维),提出了以运用能力为主要目标的集成发现教学法和教学模式。最后,进一步探讨建立以软件开发方法及其综合实践为核心的应用软件课程教学体系,重组基础课程、核心课程和相关平台应用课程关系,使这些课程内容联系更加紧密,衔接更加顺畅,有利于学生创新能力的培养。

1 新旧课程体系特点和观念变化

在 IEEE & ACM CS 2013 中,包括 18 个知识域^[2]: 算法与复杂性、体系结构与组织、计算机科学、离散结构、图形和可视化、人机接口、信息保障和安全、信息管理、智能系统、网络与通信、操作系统、基于平台的开发、并行和分布式计算、程序语言、软件开发基础、软件工程、系统基础、社会议题和专业实践。其中,软件开发基础,过去叫编程和系统导引,包括算法、设计、编程和软件开发过程。它是一些基础软件开发概念的组合作。系统基础也是一个新的知识领域,包括操作系统、网络、计算机体系结构的常用论题。为了并行计算的普适性和计算机安全更好理解的需要,加入了并行和分布式计算、信息保障和安全和基于平台的开发等课目。比以前关注较少的内容包括数字逻辑、数值方法、网页设计、搜索引擎、语言翻译等,在 CS2013 中算作选修材料。CS2013 还概括了 CS 研究生除专业技术之外与专业实践有关的一些特征,譬如沟通能力、团队合作和伦理道德,也是大学生经验的一部分。

值得关注的软件开发方法新要求,要求教学中论及一种软件开发方法时,应该考虑以下因素:开发全过程,在各阶段间的模型转换和技术衔接;便于在开发各阶段人员之间的技术交流和迭代性改进;支持有效解决问题的综合决策和管理;支持系统设计和实现的各种不同途径;支持软件正确性的检查、测试和验证;满足在系统需求中有关抽象设计、系统环境和性能的约束;支持设计师和程序员创造性的劳动;支持系统生存周期的演化;高效率使用一些自动化工具。此外,所有阶段软件开发成果物都应该是可预见和可控的。

这些新要求对软件开发的专业技术教学提出更高的标准,应该可演示、可转移、可重复验证,并且在技术实现上是开放的。

如何把计算机专业的一门软件工程学科有效地分解为初级课程和多门高级课程?如何解决现有课程体系中计算机硬件与系统课程中的裁剪和求精问题?如何在专业教学初期引入软件工程的观念,并将其在教学中与程序设计、软件职业、团队交流沟通相结合?南京大学软件学院在软件工程学科建设实践中,依据国际软件工程学科教程 CC-SE2004 建议,软件工程专业教学计划的技术课程包括初级课程、中级课程、高级课程和领域相关课程^[3],其中初级课程包括离散数学、数据结构与算法两门公共课程,另三门组成计算机科学优先方案(程序设计基础、面向对象方法和软件工程导论)和软件工程优先方案(软件工程与计算概论、软件工程与计算 II、软件工程与计算 III);中级课程覆盖计算机硬件、操作系统、网络、数据库以及其他必备的计算机硬件与计算机系统基本知识;高级课程包括 6 门课程,覆盖软件需求、体系结构、设计、构造、测试、质量、过程、管理和人机交互等;领域相关课程、与具体应用领域相关的选修课程可以结合各自学校办学特色开设。因此,在求精计算机硬件与软件系统课程的基础上,强调在专业教学的早期培养学生的软件工程观和计算机系统观,融合软件工程基础、程序设计、职业团队等知识实践的“软件工程与计算”系列课程,通过案例教授中小规模软件系统构建;围绕大中型软件系统构建知识分领域,组织软件工程高级课程;围绕软件工程应用领域,建设领域相关课程。在教材建设上编写了《软件工程与计算》三卷本教材(《软件工程的编程基础》、《软件开发的技术基础》和《团队与软件开发实践》)作为软件工程本科专业的入门课程教材。

目前专业课程教学中不同程度地存在的一些共性问题:对于教师和课程体系而言,课程之间联系不够紧,课程实践缺少系统安排,指导不够。计算思维的方法把握和指导不够;教学手段单一化,教学方法不能紧跟发展需要,教师参与和提升教学能力和水平的动力不足。对于学生而言,知识点联系缺口比较多,对自学能力不强的学生存在不少障碍;学生学习和能力提高要么盲目,要么不够主动,效率较差。抽象建模能力弱,专业思维能力有待提高;对软件工具、开发环境、软件不同阶段常用的成熟开发技术和管理手段把握不够;实际的比较复杂软件项目参与少,创新能力缺少基础。显然,对于教师、课程体系和授业对象学生,其中有些问题是相对存在的,比较突出的是课程体系集成性先天不足,不利于激发学生发现问题和解决问题的

兴趣,培养学生的专业思维能力和系统观,需要从高度集成性和问题发现、知识挖掘入手,使学生在专业素质和能力培养有较大的上升空间。

2 软件开发技术课程集成发现式教学法

2.1 软件开发中的抽象思维

计算机科学中很多涉及到算法、设计、系统的问题归结到计算机问题求解,这样思维技能的培养,可以是科学思维(例如大数据中挖掘科学规律)、算法思维(例如,根据领域原理设计和应用算法求解问题)、效率思维(计算、通信和控制体现在不同领域都有效率问题,有计算效率、经济性的效率),还有创新思维(例如,计算机网络环境下可以支持协同工作,不同专业的人员甚至普通大众可以参与到任务完成的众包问题求解,强调社会计算的思维等)。计算思维实际上不是只来源于计算机,而是更多地来自于自然和社会中人工系统设计和应用中比较共性的东西,总结出计算模型应用的模式、策略和规律来,方便交流和思考问题。从思维的角度而言,软件开发的本质就是要实现概念世界到设计模型和物理模型的映射,实现高层处理逻辑到低层处理实现的映射。抽象思维有很多思路和方法,对软件开发而言,对象思维、模式思维和架构思维很重要,三者是从微观到宏观递进的相互关联关系。

(1) 对象思维 是结构化思维的进一步发展。在结构化程序中,程序 = 数据结构 + 算法,结构化思维用“过程”来刻画数据间关系,结构化中,数据是死的,全部依赖算法操作,结构化思维更像是一个人在解决所有问题。面向对象程序,对象 = (算法 + 数据结构),程序 = (对象 + 对象 + ...)。对象思维直接用类表达数据间关系;对象思维中,数据是活的,“她”知道的信息(属性),并能完成自己的工作(操作);对象思维更像是一个团队的分工协作。当然,相对而言,对象思维是一种扬弃,不是否定结构化思维。如果说结构化编程是一种符合机器指令执行过程的一种实现,那么对象思维就是符合客观世界对象静态构成和动态交互行为的一种实现。面向对象的软件分析和设计时,不是直接面向程序代码的实现,而是需要先根据实际需求建立业务过程模型,思考应该如何去抽象描述和实现这些业务过程。从这个意义上而言,软件只是一个实际业务的载体,其中最重要的目标是业务过程的实现。统一建模语言 UML 及其可视化设计工具是这种对象思维的最好实现,作为设计活动交流手段,也支

持软件过程的迭代和改进。以用例描述和刻画需求,以多个静态视图和动态视图建立对象元素之间的联系和交互行为,使对象系统实现建立在系统分析和设计的蓝图之上,许多流行的 IDE 都支持这种集成性的实现。面向对象程序设计语言的抽象数据类型、封装性、继承性(泛化)、多态性、消息和事件机制等,保证了对象思维成果的代码级转换。

(2) 模式思维 可以认为是对象思维的高级形式。如何应用设计模式? 其实,任何复杂对象联系问题都可以通过增加一个间接层次来解决,即所谓的抽象层。设计模式的核心思维,用一句话来高度概括就是:多态包容是诸多设计模式中都反复出现的经典技术,通过多态来适应各种变化。而所谓多态包容是指宿主对象中包含抽象基类(或接口)的引用,而实际行为委托给这个引用所指向的实际对象,从而使这些行为可以根据该引用所指向的实际对象不同而不同。因为在对象级模块设计阶段,最关键的问题是系统设计的要素及其关系是经常变化的,对象设计方法上需要适应和包容这些变化^[4]。如果只是试图捕捉和描述这些变化,还是停留在分析阶段;当然,如果设计的对象模块只是能够面向未来,相对于软件开发成本和时间而言也是不允许的。实践表明,分析和设计两个阶段存在必要的转换和迭代,但过分的分析和设计都是事倍功半的。如果在分析阶段能够判断变化将要发生,到设计阶段能够判断变化将会在哪里发生,并且包容这些变化,这样就形成了几个著名的面向对象设计原则,也是对象复用设计的经验总结。例如,针对接口编程而不是针对实现编程;优先使用对象组合和聚合,而不是应用继承。考虑设计中哪些要素是可变时,什么原因导致设计改变有时并不重要,而不会引起重新设计的封装才是复用设计的重点。也就是说,封装变化的概念是模块设计的主题。掌握这一问题的思维方法,我们才需要学习重构技术影响最大而且最成功的应用,这就是 GoF 的 23 种设计模式,在 GoF 设计模式中,把设计模式分为结构型、创建型和行为型三大类,从不同的角度讨论了软件重构的方法。

(3) 架构思维 架构思维是总体宏观思维,也是高层思维。架构思维“只可意会,不可言传”,它是一门与实践紧密结合的艺术。设计模式和软件架构技术是软件复用新技术,是构件技术的发展和延伸。类库和构件库主要侧重于代码及其物理上的复用,而设计模式则侧重于软件分析和设计阶段的业务过程对象的复用。软件架构介于二者之间,它把软件的应用领域需求和特征与软件的基础架构中的要素有机地结合起来,可以认为是一种宏观的可复用构件,是一种框架技

术。主要包括三个方面的思想,一方面架构的设计起源于需求分析,另一方面架构设计更加关注软件系统的质量需求(即非功能性需求);第三方面,架构的实现还依赖于好的开发环境和项目管理。所以,软件架构设计是一个系统工程,它需要系统架构师有很宽的知识面和实施能力,从需求分析、架构设计到模块设计和实现,直到软件测试、运行和维护都需要有透彻的理解。而对于初学者而言,设计是架构、构件、接口以及系统其他特征定义的过程,需要适应企业需求和变化的架构模型元素有哪些? 如何通过分层应对复杂性和变化,构建稳定的架构? 如何通过分包,实现构件化开发? 这些问题是非常基本的。

借鉴古代三国的英雄角色,可以理解架构中的角色和功能怎么匹配的? 如图 1 所示一个 Web 应用框架^[7]。可以参照根据架构模型元素和三国故事情节之间的隐喻关系如下:单一入口相当于守卫荆州的“门神关羽”;路由相当于刘备的引路人徐庶;控制层相对于掌柜曹操;业务模型层相对于鞠躬尽瘁的劳模诸葛亮;表示层相对于战神吕布;异步处理相对于刘备的幕后英雄赵云;接口层相对于孙权的外交官鲁肃;数据层相对于曹营的后方夏侯敦;缓存层相对于曹操五个良将之一的张辽;通用类库相当于曹操的谋士郭嘉,三国演义中说“天生郭奉孝,豪杰冠群英。腹内藏经史,胸中隐甲兵。”这样阵容的精英团队组合在三国时代打天下也无敌了,那么架构的思维就是谋天下的大思维,要学会结合对象思维和模式思维,从局部到整体集成上理解架构思维。离开了架构思维,专业知识的集成和能力创新就无从谈起。

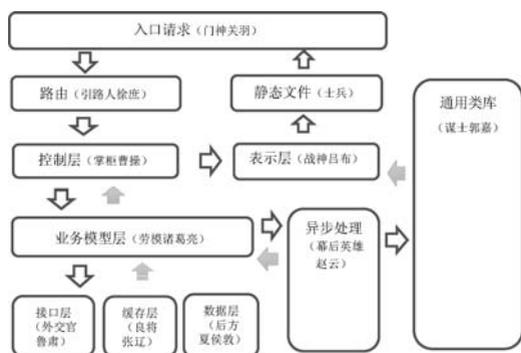


图 1 一个 Web 应用架构中的隐喻

2.2 集成发现教学法

2002 年美国学者 James F. Kurose 和 Keith W. Ross 在《计算机网络:自顶向下方法》[Computer Networking: A Top-Down Approach]一书^[5]中提出的极富盛名的“自顶向下”教学法是从计算机网络的原理及其协议教学中总结出来的,从高层的应用层协议开始沿协议栈向下层讲解,可以从实现、应用的角度更好地

理解各层网络协议的意义,教学中强调应用层范例和编程接口,学生可以在实际应用程序环境中进行学习、模仿和创造。传统的计算机课程体系人为地割裂了解决问题时所需技能的综合性。例如汇编语言、计算机原理、计算机系统结构、操作系统和编译原理分别从不同角度介绍了计算机的硬件和软件系统,但是随着多核系统日渐成为主流,这种软硬件分离的教学方法变得不切实际。国内外大学都在这方面展开了探索,即如何用一种综合的方法来讲授计算机系统的相关内容。卡内基·梅隆大学的《深入理解计算机系统》是目前比较成功的探索,它主要从程序员视角来讲解计算机系统,内容偏向系统软件(特别是操作系统)。美国佐治亚理工学院计算机系教授 Umakishore Ramachandran 提出在《计算机系统》课程实践中更多地强调采用软硬件集成的方法^[6],系统地讲解了计算机系统的软件和硬件知识及其相互关系,从发现问题入手揭示系统架构与操作系统之间的高度集成,学生才会融会贯通。例如,“什么是内存管理?”,一旦理解了内存管理的需求,再探讨内存管理所需的软件技术和相应的硬件支持,才有比较清晰的理解。计算机的内存系统可能是决定计算机性能最为关键的因素。如果内存系统不能以匹配处理器速度的方式提供执行程序所需的代码和数据,处理器速度再快也毫无意义。由于技术的进步,内存系统的大小一直在跨越式发展,但应用程序使用内存的胃口也在以同样的速度增长着。因此,内存也是宝贵资源,操作系统的作用之一就是保证用好资源。从问题发现和提出入手可以有效地集成软硬件知识,理解关于有效管理内存的操作系统算法以及相应的体系结构支持,进而把握内存层次问题,可以帮助降低处理器在访问代码和数据时的延迟。

软件开发技术是计算机科学与技术专业中最活跃、发展最快的部分,大部分计算机和软件工程的本科毕业生都从事相关的工作。当今软件开发技术中,框架、架构、面向对象分析和设计、设计模式、契约式编程、测试驱动开发、面向方面的编程、模型驱动架构、基于组件的开发、敏捷软件开发、元编程、面向服务的体系结构、面向特征的编程等许多新技术领域已经接近成熟,但据了解国内多数院校在开设和讲授软件开发技术相关课程中,都只涉及到其中部分领域,基本上没有应用自顶而下的集成发现教学方法。部分原因在于教学理念落后,教材建设滞后,教学团队建设存在先天不足,没有与行业技术专家合作教学。软件开发技术课程如何应用集成发现教学方法呢?下面以框架技术的学习为例来进行探讨。在一些著名企业招聘 Java 开发人员时,有不少经常提到的问题,以判断应聘者对

基础知识的掌握情况。例如,请解释一下 Session 的工作原理,没有很好基础的应聘者往往感到一头雾水。对于初学者而言,各种技术书籍汗牛充栋,仍然对学习各种框架疲于奔命,并不知道如何发现和解决学习中的问题,获得集成性的技术理解。如果先熟悉 Java EE 中关于 Servlet API 中常用的类和方法,在此基础上学习某个框架的使用,最后阅读 HTTP 协议的内容。对 HTTP 协议的深入了解,一旦发现和掌握了这些基础的内容,初学者就会发现使用框架甚至自己开发一个业务框架是多么容易,很自然地会引导他学习设计一个框架所用到的各种技术,其中主要包括类加载器、依赖注入、线程本地、事务管理和安全控制等。进一步地,对于初学者而言,应该学习经典的 MVC 架构 Struts 和最流行的 ORM 框架 Hibernate,并能够熟练使用 JSP + Struts + Hibernate 开发小型 Java Web 应用程序。但对于技术框架的演化而言,要适应既复杂又安全的应用需求,适应业务需求变化和 QoS 目标需求,对这些企业级应用核心需求和技术有深入的把握。所谓企业级应用,一般具有数据持久化需求、高并发访问、巨型或海量数据、应用规模庞大、界面需求复杂、生命周期长等特点,高级学习者不能仅停留在早期经典的架构技术和解决方案上。众所周知,在产业界随着 Spring 框架的问世,意味着企业级 J2EE 春天的到来, Spring + Struts + Hibernate 三个框架组合 (SSH) 被广泛应用于 Web 开发。高级学习者在实践中会发现以前的 SSH 已经显得重量级了,许多大型企业级应用会放弃掉一些技术,如 Struts、Hibernate 和 JSP 等,但 Spring 目前仍然是主流,尤其是 Spring MVC,足以取代 Struts。同时轻量级 JDBC 框架 MyBatis 也可以取代 Hibernate,目前企业级 Java EE 架构的首先变成了 Spring + Spring MVC + MyBatis,简称 SSM。

当然,对于软件开发技术的课程学习者而言,开发框架并非完整的技术解决方案,他们需要在技术方案中提出和解决虚拟文件系统、缓存、文件处理框架、模板语言、数据库访问层、数据库水平扩展、服务层、流程引擎、元数据、展现层、Web 层、统一界面框架、Restful Web 服务和模块化等满足企业级应用的核心需求。例如^[8],应对大数据量存储和高并发访问业务,需要在解决方案中考虑将应用服务和数据服务分离;为减少数据库高频并发访问的压力和服务端负荷,一般使用一些服务器缓存技术,如本地缓存、分布式缓存服务器上的远程缓存。尽管这样服务器访问压力得到一定的缓解,但单一应用服务器能够处理的请求连接是有限的,在并发访问高峰期应用服务器又成为整个系统的性能瓶颈,这时服务器集群技术的选择就非常必要;使

用缓存后大部分数据读操作访问可以不通过数据库服务器连接就能完成,但是仍有一部分缓存不命中或缓存过期的读操作和全部的写操作都需要访问数据库,在用户数达到一定规模后,服务器因为负载压力过高而又成为系统性能的瓶颈。这时在架构上有利的决策选择,可以通过配置数据库主从关系,实现数据库读写分离,从而进一步改善数据库负载压力。继而,如果从一台服务器拆分成多台服务器提供集群服务,随着应用系统业务的发展依然不能满足需求,这时就需要使用分布式数据库和大数据的系统架构。可见。解决方案中技术发现之旅,也是技术架构因需求变化而演化和重构的路径,集成和发现是一种软件技术教学的启发式途径。除此之外,还必须认识到技术和管理需要有效的结合,作为技术解决方案自然有很多可以自由选择领域,有很多受限的前提条件。这些外围因素往往还与系统背后的个人、团队、企业的价值观以及非IT能力有关。

2.3 教学模式

对从事软件开发行业的计算机专业学生而言,长期的职业发展途径大致分为三个阶段:软件工程师阶段、软件设计师阶段、架构设计师或项目管理师阶段。俗话说,不想当元帅的士兵不是好士兵,不想当架构设计师或项目管理师的程序员也不是好的程序员。从专业知识集成性和能力培养的成长路线而言,本科生期间需要学习和掌握的开发应用软件主要理论、方法和技术可以归结如下:

(A) 基础理论知识,如操作系统、计算机组成原理和体系结构、编译原理、数据结构与算法、计算机网络、信息安全等,一般的计算机相关专业培养方案和教学计划都视为核心课程。

(B) 1~2门程序设计语言,主要面向对象程序设计语言,Java/C++/C#/Python等。如果从事Web开发的话,还要学习一些网络编程技术、脚本语言、XML等。

(C) 面向对象方法学,面向对象思想、基于构件的开发CBD、面向对象编程AOP等。

(D) 关系型数据库及DBMS,例如,ORACLE/SQL Server/DB2/MySQL等。

(E) 软件开发集成开发环境和软件测试技术,例如,IntelliJ IDEA/Eclipse/MyEclipse/VS.NET等。

(F) UML建模工具,例如,Rational ROSE/VISIO。

(G) 软件过程管理,例如,RUP/XP/CMM等,通过软件过程模型和持续改进,使开发流程规范化,软件质量可控。

(H) 软件源码管理Git/GitHub、项目管理、软件架构、框架、开源软件、软件开发的经济学等。

对多数学生而言,恰当的学习路线应该是:(B)→(C)→(E)→(F)→(G)→(H)。进一步地,软件工程师着重于(B)、(C)、(E)、(D);软件设计师着重于(B)、(C)、(E)、(D)、(F);架构设计师着重于(C)、(F)、(H)。

专业教学培养体系中,软件开发课程群设置以初学者自顶向下地集成知识和发现知识、提高运用能力为导向,以软件开发方法学、工具和环境支持为教学核心。其中,软件开发方法学比较成熟的内容和做法是,从不同的软件类型,按不同的观点和原则,对软件开发中应遵循的策略、原则、步骤和必须产生的软件产物进行解构性教学,从而要求学生掌握软件开发规范化和工程化的基本技术。教学内容设置上应该覆盖软件开发模型和一般方法、软件开发平台与工具、系统分析与建模、软件架构技术、软件设计模式、IT规划与咨询、软件开发项目管理、软件开发标准和案例等内容。通过这些课程群的设置和教学实施,学生很容易掌握在操作系统、数据库、多媒体、网络等应用环境中软件的基本开发方法及技术,以及在软件开发过程中要解决的数据结构与常用算法的实现,学生在掌握了开发软件所必需的基础知识、基本开发方法和常用开发技术之后,今后面临的职业发展道路就更加有信心和目标。在同济大学计算机科学技术系的教学实践中,我们部署了如下的教学方案(要点):

1) 知识目标

(1) 教授以软件开发的发展、主要方法、常用工具的使用为主线。包括:软件开发的基本概念、基本原理、常用的软件管理方法、常用的建模工具等。

(2) 关键的知识讲授:软件需求获取方法、需求规格说明撰写与评审方法、软件设计的基本概念、结构化设计方法、体系结构设计、面向对象UML的软件分析方法、UML建模方法、构件化软件开发技术等。

(3) 通过该课程的教学活动,学生应具备软件开发基本理论知识。

2) 能力目标

(1) 运用软件开发的观念、原理、技术和方法开发与维护软件的能力。

(2) 编写相关技术文档的能力。

(3) 分析问题和解决问题的能力 and 团队工作交流能力。

3) 素质目标

(1) 职业素养的培养:对软件开发领域常见的职业要求、团队合作精神进行培养与灌输。

(2) 专业素养的培养:对软件管理过程中常见的问题能够予以准确判断,明确解决思路。

4) 教学方案及其改进要点

结合 IEEE & ACM CS 2013 课程体系,2014 年我校计算机科学与技术专业又一次修订培养计划^[9],并开始制定课程教学大纲和实验教学大纲,进一步完善教学体系、教学团队相关支持。其中,软件技术系列的课程包括原有的:高级语言程序设计、汇编语言程序设计、数据结构与算法、数据库系统、操作系统、编译原理、软件工程、程序设计方法、服务计算概论等,又增加了计算机系统实验(必修)、软件开发方法(必修)、Unix 系统分析(选修)、软件形式化技术(选修)等课程。这也体现了目前国际上计算机科学学科知识体系在系统基础和软件开发基础方面有了较大的变化,其中,前者对应一个新的知识领域,包括操作系统、网络、计算机体系结构的常用论题,强调系统思维方法和系统能力的提升。而后者过去主要依赖前期程序设计、数据结构和算法课程以及后期的软件工程课程,按照 CS2013 的观点,算法、设计、编程和软件开发过程是一些基础软件开发概念和技术的组合,其中软件建模和基于平台的软件开发是信息技术应用和创新的最重要基础部分。

在计划调整过程中,以培养和提高学生复杂软件系统的构建和实现能力以及新技术应用能力为目标,建立软件开发全局性的目标矩阵和技术能力矩阵,各门课程的建设相互联系、相互支持。例如:

(1) “软件开发方法”中构件技术、设计模式以及流行的软件开发环境和工具,与“高级语言程序设计”课程中面向对象程序设计、“软件工程”中面向对象的分析与建模,教学上联系起来,培养学生软件开发的分析、设计建模和实现能力;程序设计是软件架构实现的核心,通过代码可以完成对现实世界的虚拟化:概念的虚拟化、能力的虚拟化、实体的虚拟化、记忆的虚拟化、协作的虚拟化,例如,概念的虚拟化:对应一个业务概念的类定义;能力的虚拟化:一个方法对多个输入数据进行加工并返回结果;实体的虚拟化:一个类的实例,即具体的数据;记忆的虚拟化:一条关系型数据库的行记录;协作的虚拟化:远程方法调用。程序代码是计算机的指挥者,通过代码把人类智慧「赋能」给计算机的一种语言。对象设计模式主要支持功能性需求;而架构模式支持非功能性需求。现代软件架构设计的原则来自于软件复用,软件复用是指重复使用“为了复用目的而设计的软件”的过程。

(2) “程序设计方法”与“软件形式化技术”、“软件可靠性与容错”与“软件测试”课程联系起来,培养

学生软件设计验证、性能优化与质量保证能力。

(3) “软件开发方法”课程中将软件体系结构与“服务计算概论”中的面向服务架构联系起来,将软件建模与“电子商务技术”中业务建模联系起来,培养学生对复杂软件架构和软件开发环境的应用能力和创新基础。

(4) 将先期的软件技术课程和后期的“数据库系统原理”、“计算机网络”、“Web 技术”、“软件工程”课程教学联系起来,培养学生软件过程改进与迭代、软件开发组织、管理和沟通能力。

(5) 适当注意本科阶段与研究生阶段课程与实践在深度和广度上的衔接和联系。

2014 年培养计划和教学方案调整基本体现了集成发现教学法的精神。在软件开发技术系列课程建设上,试图建立从程序设计到软件工程之间一系列课程的连接桥梁。课程的建设标准以工程专业认证通用标准(12 条)为基准,达成预定的技术能力矩阵,并在教学中强调课程的毕业达成度评价和持续改进。在课程实践环节,聘请校外技术专家参与教学和指导环节。

3 结 语

本文从计算机专业中软件开发技术课程设置和教学改革的要求出发,分析了新旧教学体系的特点和目前存在的主要问题。基于软件开发实践中的抽象思维(对象思维、模式思维和架构思维),提出了一种适合于软件开发技术课程的集成发现教学法和教学模式。本文探讨了如何建立以软件开发方法及其综合实践为核心的应用软件课程教学体系,如何引导和支持学生自主地发现所学知识,提高应用实践的集成运用能力,是需要借助于很好的教学设计、合理的教学计划支持。

今后需要重点关注的是,创新和创业已经成为现阶段我国经济转型和企业升级的重要发展策略,而技术创新驱动商业模式创新已成为 IT 创新的一个重要范式。软件开发方法创新能力的培养起点高、难度大,有必要重点研究软件新技术的教学实践。例如,现在从事企业软件开发,掌握企业信息系统软件开发中构件技术的应用很有好处。目前电商系统架构、大规模信息服务架构、大数据应用架构、云计算架构、物联网架构都有应用,教学实践中一方面重点介绍成熟的构件技术及典型的分布式架构技术,但作为创新的基础训练,可以结合一些新的选题,提供给学生足够的学习支撑和技术支持,例如,开源软件架构、网构软件及其自适应演化技术、面向方面的软件开发、高安全的形式化验证、面向 Agent 的软件开发、敏捷软件开发、面向移动应用 App 开发、包管理系统 Docker 与分布式服

务框架、云件开发等,引导学生创新的思维和计划实施能力的提高。希望这部分的内容可以根据实际需要和可能在课程项目、课程设计、专业实习和毕业设计期间有序地安排和实施。

参 考 文 献

- [1] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science[M]. ACM, 2013.
- [2] 闵应骅. 计算机科学课目 CS2013[OL]. <http://blog.sciencenet.cn/blog-290937-803714.html>.
- [3] 骆斌, 丁二玉, 刘钦. 软件工程与计算(卷2): 软件开发的技术基础[M]. 北京: 机械工业出版社, 2012.
- [4] 谢新华. 软件架构设计的思想与模式[OL]. 中科院计算所培训中心, 高级软件系统架构师培训. <http://www.TCICT.cn>.
- [5] Behrouz A Forouzan, Firouz Mosharraf. 计算机网络教程: 自顶向下方法[M]. 陈鸣, 译. 6 版. 北京: 机械工业出版社, 2013.
- [6] Umakishore Ramachandran, William D Leahy. 计算机系统: 系统架构与操作系统的高度集成[M]. 陈文允, 译. 北京: 机械工业出版社, 2015.
- [7] 应用 Web 框架模块设计三国演义篇[OL]. 2015-09-16. <http://weibo.com/p/1001603887721239202407>.
- [8] 大型网站架构演化历程[OL]. 2015-06-03. <http://blog.csdn.net/zhaoxuejie/article/details/46347063>.
- [9] 同济大学计算机科学与技术本科生专业培养计划(2014 版)[OL]. <http://see.tongji.edu.cn/09/fb/c319a2555/page.htm>.

(上接第 13 页)

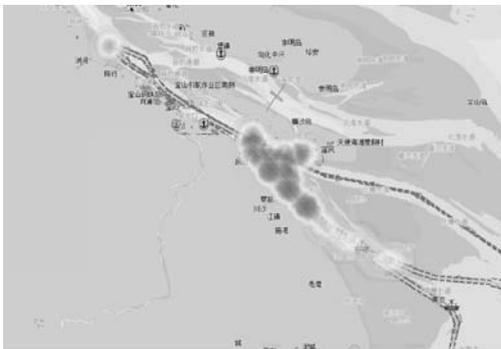


图 9 高碰撞风险区域热图

5 结 语

本文所研究的碰撞风险识别系统,通过 AIS 数据

提取船舶邻域,利用改进聚类算法,识别船舶碰撞风险,具有一定的智能性,可以主动识别船舶碰撞风险,与 VTS 系统需要人根据经验和实际情况去判断相比,具有一定的优越性。通过研究发现,构建的碰撞风险识别系统具有现实可行性,可应用于辅助决策,它可以辅助海事主管机关对辖区水域的船舶交通形势进行监管,根据识别出的碰撞风险情况,提前向相关的船舶发出预警,保障船舶交通安全,适当减轻了海事监管人员的工作强度,提升了监管水平。系统中集成的区域、多船船舶轨迹回放功能,可将事故现场重现,为水上交通事故调查提供参考;模拟航行功能可预测特殊船舶在航道航行中可能遇到的窘迫局面,提醒提醒海事监管人员关注引导,避免窘迫局面的产生。在系统投入应用以后,处理险情超过 80%,实际的使用过程中,系统但还存在一定的误警率,在未来的研究过程中,可以通过历史 AIS 数据和事故情况构建训练集、测试集,运用机器学习的相关方法,提高系统的识别准确率。

参 考 文 献

- [1] 蔡诚君. 基于场论的水域船舶碰撞危险度研究[D]. 大连海事大学, 2015.
- [2] 周梦婕, 葛泉波, 黄习刚, 等. 海上智能交通系统的新型船舶航行系统风险识别方案[J]. 中国航海, 2013(4): 114-118.
- [3] 李丽娜, 王俊玲, 陈国权. 船舶拟人智能避碰决策理论的集成机器学习策略[J]. 信息与控制, 2011, 40(3): 359-368.
- [4] Liu Z, Hu M B, Jiang R, et al. Method to enhance traffic capacity for scale-free networks[J]. Physical Review E, 2007, 76(3): 037101.
- [5] 杨鹏, 邹时林. 基于 OpenLayers 的 WebGIS 客户端的研发[J]. 测绘与空间地理信息, 2012, 35(3): 131-133.
- [6] Tang Y, Sun H, Wang X, et al. An efficient and highly available framework of data recency enhancement for eventually consistent data stores[J]. Frontiers of Computer Science, 2017, 11(1): 1-17.
- [7] 王博千, 于齐, 刘辛, 等. 面向 Cassandra 数据库的高效动态数据管理机制[J]. 计算机科学, 2016, 43(7): 197-202.
- [8] Hansen M G, Jensen T K, Lehn-Schiöfler T, et al. Empirical ship domain based on AIS data[J]. Journal of Navigation, 2013, 66(6): 931-940.
- [9] Yan Z L, Luo W J, Bu C Y, et al. Clustering spatial data by the neighbors intersection and the density difference[C]// Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies. ACM, 2016: 217-226.