

一种基于知识点结构的分布式协同学习一致性模型分析

王晓华 李天泽

(西安电子科技大学网络与继续教育学院 西安 710071)

摘要 现代远程教育是基于互联网和终端实现教学活动,协同学习是提高远程学习效能感的重要方式,需要深入研究基于协同学习语义的一致性维护、协同感知等关键技术,以达到足够高效的、自然的互动化及个性化学习。提出一种远程教育特点下协同学习的一致性模型及算法分析,在此模型基础上,形成基于知识点结构的特有的操作转换算法来保持一致性。

关键词 CSCL, 知识点, 操作转换, 分布式实时系统

Consistency Model of Distributed Collaborative Learning Based Knowledge Point

WANG Xiao-hua LI Tian-ze

(Network and Further Education College, Xi'an University of Electronic Science and Technology, Xi'an 710071, China)

Abstract Modern distance education is based on the Internet and terminals with teaching activities, collaborative learning is an important way to improve the efficacy of distance learning, it is required that in-depth research of consistency maintenance, collaborative awareness and other key technologies based on collaborative learning semantic, so high efficient, natural interaction and customized learning will be achieved. This paper analysed model and algorithm of collaborative learning consistency about the distance education. By this model, a specific algorithms based on knowledge point structure would be Build to maintain the consistency.

Keywords CSCL, Knowledge points, Operational transformation, Consistency

1 引言

现代远程教育是基于互联网和终端实现教学活动,教学中实时的异地的虚拟环境主要有远程课堂教学、辅导答疑、小组学习讨论等,必须通过互动的形式实现教师与学生无障碍的、流畅的教导和学习,并且协同学习是提高远程学习效能感的重要方式。但是制约协同学习发展与应用的主要因素是配套的协同学习系统(CSCL)在协同学习模型、协同学习感知、个性化副本一致性、基于协同学习语义的并发控制等关键技术还没有深入的研究,以达到足够高效的、自然的互动化、个性化学习。需要针对分布式实时协同学习系统在现代远程教育的应用背景下主要关键算法进行研究,本文提出一种远程教育特点下协同学习的一致性分析及算法。通过改进数据结构,形成基于知识点的特有的操作转换算法来保持一致性,并针对不同角色进行一致性副本的非对称映射。基于实时协同编辑系统的协同学习系统一般采用半复制的、分布式结构,那么协同学习系统的一致性问题非常重要,它的解决方法与应用语义相关,目前针对不同类型的协同编辑系统,已提出了多种一致性模型和算法,典型的有操作转换、多版本控制、锁方法等。对于远程教育,基于知识点的学习模式需要设计新的知识点数据结构,并在已有的一致性方法基础上形成基于新的数据结构的一致性维持算法,主要针对歧义、因果冲突、意

图冲突 3 类进行研究。特别是协同学习中复杂对象、基于知识点的数据如何在实时的协同学习中保持一致性,这需要在已有的协同一致性算法上改进发展。首先设计新的基于知识点的数据结构,其次改进锁机制处理冲突,提出协同学习中特有的操作转换算法来保持一致性。本文主要探讨协同学习中基于知识点结构的一致性的模型。

2 一致性模型及算法

2.1 模型

基于知识点的复杂对象从表现形式分类,包括文本、图形、图像、视频、音频、嵌入的各类对象(PPT、数据表、xml等)、各类数据库等,故在分布式虚拟实时协同学习中需建立新的多维的数据体系结构和模型,以适应广域网环境下高效的协同学习、协同感知等目标。广域网环境下协同学习的特点如下。

1. 网络环境不稳定,会产生拥堵、抖动等问题,导致实时系统面临更多的响应时间难点。
2. 获取资源受限,由于广域网环境,故学习资源不能够对等的传递给所有学生,只能进行适应性的提取和映射。
3. 学生使用的异构平台,必须考虑兼容性。
4. 协同感知困难,必须从知识点的结构上提取出语义层描述。

本文受西安电子科技大学基本科研业务费资助。

王晓华 博士生,讲师,主要研究方向为计算机协同学习、网络教育、计算机体系结构等,E-mail: Xh_wang888@163.com;李天泽 硕士,工程师,主要研究方向为教育技术学、软件工程等。

5. 调整知识点的模型和描述,以支持动态的、分布式的数据存储方式。

6. 一致性和并发的操作粒度分层实现。

针对以上特点,基于知识点结构广域网环境下协同学习的一致性模型如下。

2.2 知识点的描述

在协同学习过程中常把知识点可划分为元知识点、知识点、知识单元(课程、主题讨论、协同作业、主题答疑等)、知识领域(专业、交叉专业等)、学科。在一定知识支持下不能再分割的框架结构的知识点称为元知识点,比如一段文字、音频文件等,它是教学过程中最基本的教学单元。元知识点在教学过程中是不可划分的,必须作为一个整体进行教授。由两个或两个以上的元知识点组成的知识点称为知识点,也可以看作是一个知识块,其在教学过程中是一个逻辑整体。知识点划分的粒度要适宜,以不影响学生理解问题为原则。在一个课程体系或知识领域中,元知识点是确定的,而知识点则不完全确定。知识点可以根据授课教师的经验动态将其转换成语义约束,从而自适应地构成^[1]。如图1的层结构的知识体系中,任何一个下层实体都有至少一条有向边指向其上层实体,任何一个上层实体都有至少一条有向边指向其下层实体,故上层实体和下层实体是一对多、多对一、多对多的关系。知识点表示为一个九元组(ID, Name, Description, Keywords, Type, Applicability, Importance, Difficulty, Extension)。其中ID为知识点的标志号,由两部分组成:创建者+时间戳;Name为知识点的名称,即元知识点的标题;Description为知识点的描述部,以文字或其他媒体的形式对知识点的内容做一个概要的描述,可以是某文件的链接,指向内容的实际存储位置;Keywords为知识点的关键字集,即可用于检索本知识点的关键字集;Type为知识点的类别,值域为{父结点,前导点集,参考点集,支持点集,子结点集,阈值},学习本知识点后的测试成绩不小于这个阈值,才可以离开本知识点,其中父结点为空表示是根知识点(可以不是知识领域学科等),子结点集表示是元知识点。Applicability是知识点的适用范围,表明该知识点适合于哪类学生学习;Importance是知识点的重要程度,体现该知识点重要程度,取值范围是[0,1],值越大表示该知识点越重要;Difficulty是知识点的难度,体现该知识点的难易程度,取值范围为[0,1],值越大表示难度越大;Extension是知识点的发布者对知识点的可扩充内容,如专业教师富有创意的知识点说明。

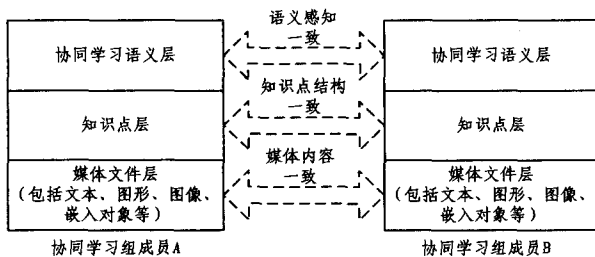


图1 一致性模型

2.3 知识点结构及操作分析

分析协同学习的过程,初始的个性化课程知识体系为某一事先确定好的模板(或发起者比如教师或学习组长等,根据某一原则静态生成),在学习的过程中小组所有成员动态构建

该知识体系,也就是协同学习。那么对知识点的操作一般可以有查看、插入、删除、创建、编辑、替换、合并、拆分、练习。其中查看、插入、删除、创建、编辑为基本操作,其他为复合操作,复合操作可以由若干基本操作形成。创建操作中元知识点可以继承空元知识点也可以继承已有的知识点,并修改其属性(不包括存储媒体文件)。编辑操作是修改元知识点中具体的媒体文件(文本、图形、图像等),需要考虑协同编辑的并发和一致性维护。比如某个画图作业,需要小组成员共同完成,由于元知识点内容以不同媒体类型存储,因此具体媒体的协同编辑则根据媒体类型选择相关算法保持一致性。特别指出复合操作“练习”的含义:查看某一特定元知识点(考题、作业等本体)后,协同或个体完成习题,形成新的学习记录(通过创建新知识点来记录)。

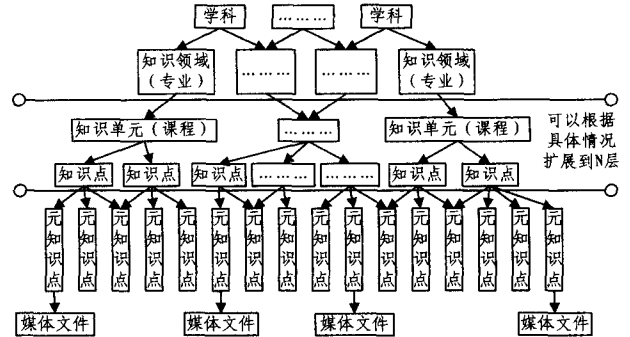


图2 知识点树形结构

知识点可以由若干知识点聚合而成,知识结构中的各知识点之间形成树型结构,如图2所示。知识点的层次关系是按照横向结构与纵向结构对知识的划分而得到的。根据纵向和横向划分的不同知识点之间的关系可将其分为父子关系和兄弟关系。(1)父子关系(包涵关系)也称作辈分关系。在知识树中具有父子关系的两个节点是父子关系知识点。父子关系是传递的。如果两个知识点A和B具备父子关系(A是父结点),那么在合适的知识点访问序列中,B将出现在A的前面,但反过来未必成立。(2)兄弟关系。在知识树中如果某些子节点有同一父知识点,则这些子节点之间的关系是一种并列或同级关系,各知识点间为兄弟关系。如果在兄弟知识点之间不存在依赖关系,那么这两个知识点也可以称为知识点平行关系,知识点平行关系具有对称性和传递性。(3)知识点依赖关系:知识点在学习过程中具有一种必然的先后关系。一个知识点是否可以进行学习往往需要先学习另一些知识点,或者说后者是前者的预备知识。知识点依赖关系也被称为顺序关系,但A知识点未必是B知识点的组成部分。知识点依赖关系具有传递性,如果在学习完知识点A后可以直接学习知识点C,则知识点C直接依赖于知识点A。如果在学习完知识点A后,还需要再学习其他知识点才能学习知识点C,则知识点C间接依赖于知识点A,同时知识点依赖关系也可以由另一种表述,即前驱关系和后继关系。在学习某一知识点之前必须先学习相关的另一知识点,这两者之间的关系即为前驱关系。在学习某一知识点之后,由本知识点直接支持的知识点,这两者之间就直接构成了后继关系。(4)参考关系:在一个课程体系或知识领域中通常会有很多关联知识,这些关联知识之间存在部分或全部相同的内容重点,但他们之间却不一定存在依赖关系或包含关系,这时称他们之间

的关系为一种参考关系。参考关系满足对称性,但参考关系未必是传递的。(5)游离关系:如果知识点 A 属于某领域知识树,知识点 B 不属于这个领域知识树,但知识点 B 能丰富知识点 A 的学习,这时 A、B 之间是一种游离关系。游离关系类似于参考关系,不同的是参考关系建立在课程知识体系或知识领域之内,而游离关系关联的是课程知识体系之外的知识点。游离关系是不同领域知识点之间的关系,而其他 4 种关系都是同一领域知识点间的关系^[3]。

2.4 知识点体系结构一致性分析

基于知识点的操作转换算法基于一种全序关系来进行调度,故先引出各类操作关系。

定义 1(因果关系) 给定任意两个分别位于站点 I 和站点 J 上的操作 O_a 和 O_b ,称 O_a 和 O_b 存在因果关系(记做 $O_a \rightarrow O_b$),当且仅当 O_a 和 O_b 满足下列 3 个条件之一:① $I=J$ 并且操作 O_a 发生在 O_b 之前;② $I \neq J$ 并且操作 O_a 在站点 j 的执行先于操作 O_b 的产生;③存在操作 O_x ,并且有 $O_a \rightarrow O_x$ 和 $O_x \rightarrow O_b$ 。

定义 2(依赖关系/平行关系) 给定任意两个操作 O_a 和 O_b ,称 O_a 和 O_b 具备依赖关系,当且仅当 O_a 和 O_b 满足 $O_a \rightarrow O_b$ 或者 $O_b \rightarrow O_a$;称 O_a 和 O_b 具备非依赖关系(平行关系,记做 $O_a \parallel O_b$),当且仅当 O_a 和 O_b 既不满足 $O_a \rightarrow O_b$,又不满足 $O_b \rightarrow O_a$ 。定义 2 实际上是定义 1 的另一种表述方式,对于存在依赖关系的操作可以按照因果次序来定序,对于具有非依赖关系的操作还可进一步分类(与知识点的依赖关系/平行关系不一样,此处是指操作的依赖关系/平行关系)。

定义 3(知识点体系结构一致性模型) 如果满足下列 3 个条件,则协同学习知识点体系是一致的:1. 最终结果一致:在执行完相同操作集后,所有站点的共享知识点体系副本是一致的;2. 因果保持:任意两个操作 O_a 和 O_b ,如果 $O_a \rightarrow O_b$,那么所有站点 O_a 先于 O_b 执行;3. 意图保持:对任意一个操作 O , O 在本地和其他站点执行效果与 O 的意图相同,并且,如果存在 O_x 且 $O_x \parallel O$,那么 O_x 的执行效果与 O 的执行效果互不影响。

定义 4(时间向量) 设 N 为协同学习中的站点数,各站点的标号分别为 $0, \dots, N$ 。每个站点的时间状态向量 SV 都具有 N 个分量。初始时, $SV[i]=0, i \in \{0, \dots, N-1\}$ 。站点 i 执行一个操作后, $SV[i]:=SV[i]+1$ 。任何操作在本地站点执行后,将操作及本地的时间状态向量 SV 发送到远程协同学习站点。

定义 5(全局序关系“>”) 给定两个分别由站点 I 和 J 产生的操作, SV_1 和 SV_2 是相应的时间向量。 O_1 在全局序中先于 O_2 , 记为 $O_1 > O_2$, 当且仅当: $(Sum(SV_1) < Sum(SV_2))$ 或者 $(I < J$ 且 $Sum(SV_1) = Sum(SV_2)$), 其中 $Sum(SV_j) = \sum_{i=1}^n SV_j[i]$ 。对于任意两个操作 O_1, O_2 , 如果 $O_1 > O_2$, 则 O_1 是 O_2 的前序操作, O_2 是 O_1 的后序操作。对所有操作可以确定唯一的全局序关系, 全局序和因果序是一致的。如果各站点操作按照操作的全局序执行, 或者对于有冲突的操作做转换处理使得操作执行效果等价于按全局序执行的效果, 那么协同学习系统的最终知识点体系结果是一致的, 符合定义 3 的 3 个一致性要求。据此, 我们提出一种新的基于知识点的操作转换的算法, 通过对冲突操作做执行前的操作转换(冲突一般包括父子关系冲突、依赖关系冲突、同一知识点

操作冲突等), 然后执行转换后的操作, 从而实现协同学习系统知识点体系结构的一致性。

2.5 知识点体系结构一致性及操作转换算法

定义 6(背景知识点结构 BC) 操作 O 产生时的知识点结构称为操作 O 的背景知识点结构, 记为 $BC[O]$ 。背景知识点结构是知识点操作的上下文, 记录了当前知识点结构中所有知识点的属性和关系, 本文讨论的知识点结构基于树状结构。初始知识点结构记为 BC_0 。操作 O 在背景知识点结构上执行后得到的知识点结构为结果知识点结构, 记为 $BR[O]$ 。对于任意一个操作 O , 显然有 $BR[O] = BC[O] + O$, “+”表示 O 作用在 $BC[O]$ 上。对一组操作: $HB = [O_1, O_2, \dots, O_n]$, 有 $BR[O_n] = BC_0 + O_1 + O_2 + \dots + O_n$ 。

定义 7(背景知识点结构等价关系) 对于产生于两个站点的操作: O_1, O_2 , 如果 $BC[O_1] = BC[O_2]$, 则 O_1 和 O_2 的背景知识点结构等价, 表示为 $O_1 \equiv O_2$ 。

已执行操作按照全局序保存在历史队列 HB 中, 一个站点已执行操作 O_1, O_2, \dots, O_n , 记为 $HB = [O_1, O_2, \dots, O_n]$, 且有 $O_1 > O_2 > \dots > O_n$ 。知识点操作 O 可以用 $Costudy(PS)$ 表示, PS 表示在背景知识点结构 BC 上查看、插入、删除、创建、编辑元知识点。在协同学习系统中, 一般发起了一个协同学习任务后, 所有学习者在终端上通过点击知识点链接, 进入具体的知识点内容中来完成协同学习。每个终端后台维护一份背景知识点结构树副本, 对于本地操作, 在操作还没有完成前, 操作结果不写入背景知识点结构, 而是暂存在临时操作中, 直到操作完成才调用 $sIOPT$ 算法做操作转换和更新背景知识点结构。对于远程操作, 收到后直接调用 $sIOPT$ 算法做操作转换和更新背景知识点结构; 由于因果关系和全局序是一致的, 因此利用操作转换, 因果关系操作可以不按照因果序执行, 也可以实现因果保持的执行效果。

定义 8(运算 \oplus) 插入知识点, 用 $Insert(A, B)$ 表示。其中 B 为父知识点。将知识点 A 插入到知识点结构树中, 作为 B 的子结点, 形成知识点间的包含关系。在知识点 B 的子结点集中添加 A , 在知识点 A 的父结点中添加 B 。

定义 9(运算 $-$) 删除知识点, 用 $Delete(A)$ 表示。首先确定知识点 A 是否有子知识点且是否是其它知识点的前导知识点。只有没有子知识点的知识点才可以删除, 可以通过遍历知识体系树得到知识点 A 的所有子知识点和后续知识点。如果知识点 A 删除后, 知识树中所有后续知识点与 A 相关的前导属性也删除, 同时 A 的所有父知识点的子知识点属性集中 A 被删除;

定义 10(运算 \otimes) 前驱知识点, 用 $Link(A, B)$ 表示。指定知识点 A 必须在知识点 B 前学习, 亦知识点 B 是知识点 A 的后续知识点。在知识点 B 的前导知识点集中添加 A 。对 B 为根节点时需单独考虑。

定义 11(运算 \yenarrow) 创建知识点, 用 $Create(A)$ 表示。新知识点 $B = Create(A)$, 其中 A 为已有的知识点, B 为生成后的新知识点, A 可以为 ϕ , B 的所有属性和 A 一样。

定义 12(运算 \cup) 合并知识点, 用 $Merge(A, B)$ 表示。合并有 3 种情况: B 为根知识点; 直接父子知识点; 直接兄弟知识点。合并操作后, 必须确保其余知识点与新知识点间的关系不发生变化, 新知识点必须继承知识点与其它知识点所

(下转第 256 页)

- [5] Thomas Sandholm and Kevin Lai, Dynamic proportional share scheduling in hadoop[C]//JSSPP '10: 15th Workshop on Job Scheduling Strategies for Parallel Processing, 2010; 110-131
- [6] Polo J, Carrera D, Becerra Y, Performance-driven task co-scheduling for mapreduce environments[C]//Network Operations and Management Symposium(NOMS), IEEE, 2010; 373-380
- [7] Tian C, Zhou H, Zha L, A dynamic MapReduce scheduler for heterogeneous workloads[C]//Proceedings of the 2009 Eighth International Conference on Grid and Cooperative Computing, IEEE Computer Society, 2009; 218-224

- [8] 陈全, 邓倩妮. 异构环境下自适应的 MapReduce 调度[J]. 计算机工程与科学, 2009; 168-171
- [9] 孟令芬. pc 集群作业调度算法研究[D]. 东营: 中国石油大学(华东), 2009
- [10] Apache Hadoop[OL]. <http://hadoop.apache.org/>
- [11] Fair Scheduler for Hadoop[EB/OL]. http://Hadoop.apache.org/common/docs/current/Fair_scheduler.html, 2010-04-15
- [12] Capacity Scheduler for Hadoop[EB/OL]. http://Hadoop.apache.org/common/docs/current/Capacity_scheduler.html, 2010-03-22

(上接第 241 页)

有的知识点关系。依据两知识点问所处位置的不同, 有 3 种合并形式。

1) 合并直接兄弟知识点: 表示将知识点 B 的属性合并到知识点 A 上。由于多个知识点的合并可以简化为若干次两个知识点的合并过程, 因而这里仅讨论两知识点合并中知识关系的变化过程。合并直接兄弟知识点要求合并两知识点的知识关系, 将 B 的前导知识点集、具体知识内容添加到 A 的对应属性中, 然后删除 B 。

2) 合并直接父子知识点: 仅在父知识点 A 有唯一的子知识点 B 且子知识点 B 有唯一父知识点 A 时, 方可进行合并。父子知识点合并后, A 拥有 B 所有的关联关系和父子关系, 且在 A 的包含属性集中需删除与 B 有关的包含属性值, 然后删除 B 。

3) 合并根知识点: 仅 B 为根知识点时方允许合并。设知识点 A 是知识点 B 的前导知识, 知识点 B 是知识点 A 的后续知识。根知识点合并后, A 拥有 B 所有的关联关系和父子关系, 然后删除 B ^[5]。

定义 13(运算 \cap) 拆分知识点, 用 $Split(A, B)$ 表示。将知识点 A 分裂为 A, B , 知识点 A 的各种属性点集中删除 B 相同的属性点集, A 作为 B 的父知识点, B 作为 A 的子知识, A 和 B 中具体的知识内容由组成员协商。

定义 14(运算 \S) 查看知识点, 用 $Look(A, B)$ 表示。必须已查看 A 的全部前导知识并达到指定学习阈值才可以查看 A 。首先, 产生新知识点 $B = Create(A)$, 写入达到学习阈值(比如成绩), 然后 $Insert(B, A)$, 形成学习的历史知识点记录。

定义 15(运算 $<$) 确定全局序关系: 用 $Global(Oa, Ob)$ 表示。确定操作 Oa 和 Ob 的全局序关系。

算法 1 sIOPT 算法

```
//HB 中有 n 个已执行知识点操作
HB:  $O_1, O_2, \dots, O_n$ 
 $Ok_1$ : 该站点收到的远程操作
BC: 站点的当前背景知识点结构  $BC = BR[On]$ 
sIOPT( $Ok_1$ ) {
//封装  $Ok_1$  操作的所属知识点操作, Package( $Ok_1$ ) 中进行具体媒体文件的操作一致性处理, 本文暂不讨论。
 $Ok_1 = Package(Ok_1)$ 
//  $Ok_1$  是  $On$  的后序操作
If  $Ok_1 < On$  Then
//  $Ok_1$  加到 HB 尾
```

```
HB.AddTail( $Ok_1$ )
//执行  $Ok_1$  更新背景知识点结构
 $BC = BC + Ok_1$ 
Return
End If
 $j = n$ 
// $Ok_1$  是的  $O_j$  前序操作
While ( $O_j < Ok_1$ )
 $j--$ 
End While
 $Ok_1' = Ok_1$  // 对  $Ok_1$  所有后序操作做包含转换
For  $m = j + 1$  To  $n$ 
 $Ok_1' = IT(Ok_1', Om)$ 
End For
//将  $Ok_1$  插入 HB 中  $O_{j+1}$  的前面
HB.InsertBefore( $Ok_1, O_{j+1}$ )
//更新当前背景知识点结构, 显示执行效果
 $BC = BC + Ok_1'$ 
}
```

定义 16(知识点操作转换 $IT, O1, O2$) 对于操作 $O1, O2$, 如果 $O1 > O2$, 且 $O1 \approx O2$, $O1$ 对 $O2$ 做包含转换 $O1' = IT(O1, O2)$, 使得执行 $O2, O1'$ 和执行 $O1, O2$ 的结果一致, 即: $BC[O1] + O1 + O2 = BC[O2] + O2 + O1'$ ^[6]。

结束语 本文解决了基于知识点树结构的协同学习中, 组成员协同完成一个学习任务的过程中如何进行知识树及具体的媒体文件的一致性维护, 提出了一种基于知识点操作转换的一致性算法模型。为后续的协同学习相关研究奠定了基础。

参考文献

- [1] 贾巍, 程莹. 知识表示视野下网络课程知识点关系研究[J]. 继续教育研究, 2008
- [2] 张平安. 网络课程中的知识结构化表示方法[J]. 计算机教育, 2008
- [3] 朱郑州. 基于知识点本体的个性化课程组织方法[J]. 计算机科学, 2009, 36(12)
- [4] 廖斌. 实时协同工作系统中操作转换算法综述[J]. 计算机研究与发展, 2007, 44(2)
- [5] 肖锐. 课件知识点操作的形式化研究[J]. 计算机科学, 2008, 35(1)
- [6] 徐向华. 可适应的实时协同编辑系统若干问题研究[D]. 2005