



第 3 章

Person-to-Person 过程： 计算机支持的协同工作

CLARENCE A. ELLIS, PAULO BARTHELMESS, JUN CHEN, and JACQUES WAINER

3.1 引言

支持人们进行沟通和协作的计算机系统,是辅助“计算机支持的协同工作”(Computer-supported Collaborative Work, CSCW)的重要工具,隐含着人员交互的过程。本章将对这些系统和过程的相关问题进行讨论:这些系统和过程聚焦于信息系统中的社会技术问题,因此显得尤为重要;越来越多的人参与到信息系统之中,因此信息系统中人员协作的问题也愈加重要;人员交互过程所具有的复杂性、半结构化和动态性,使得对这些过程的计算机化更具挑战性;人们经常需要针对信息系统进行异常处理和动态变更,使得这类系统显得更加重要。因而,有必要对影响组织结构、社会环境和文化氛围的因素加以考虑。

本章将介绍人与人(Person-to-person, P2P)的过程及其支持系统。这类系统称为过程感知的信息系统,它隐式表达了人员交互的过程。首先,我们讨论协同工作的特征、所隐含的协同技术、存在的问题和误区。其次,引入一些用于描述 P2P 系统的概念和术语,并用它们对 5 个 P2P 系统进行刻画和对比。最后是总结、练习和参考文献。

3.2 人与人交互的特征

本节将讨论协同工作的相关问题,通过实例介绍协同工作中交互的复杂性和多变性,并讨论如何识别这些交互的模式。最后,探讨社会技术的内涵及其存在的问题和误区。

3.2.1 协同工作

“人们如何工作是保守得最好的秘密之一”(David Wellman, 在文献[44]中被 Suchman 引用),这在今天仍不失为一个真理。协同工作的特点在于由诸多因素相交织而成的复杂性和多变性,这些因素包括组织、社会、政治、文化以及感情等。工作中的交互形式多种多样,例如:

- 极限协作

Mark^[28]描述了 NASA 的喷气推进实验室(Jet Propulsion Laboratory, JPL)所采用的“作战室(war room)”方式: 16 位专家被集中在一个配有网络和公共显示器的房间里工作, 要求他们在极短的时间内, 为复杂的太空任务设计出一套完整详尽的任务方案。

在这种方式下, 合作通过感知的复杂组合实现: 监听嘈杂环境中的谈话、对共享数据做出响应(这些数据通过定制的关联电子表格传递, 团队成员可以发布自己的数据, 也可以订阅他人发布的数据)。小组成员可以走访房间内的其他专家, 或者聚集到公共显示器前对感兴趣的问题进行讨论。他们的行动会向房间内的其他人传递感知信息, 其他人可以选择是否加入讨论(这取决于自己的工作与参与讨论的专家的相关性)。在工作中, 团队成员会意识到周围的谈话, 对相关的关键词做出响应, 如在自己的座位上给出简短的回答或直接起身加入到讨论中。类似地, 电子表格系统提供的数据也能引发协作。

- 国会会议

根据 Robert 在《罗伯特议事规则》^[33]中的描述, 国会会议属于非常正式的交流。参与者具有非常明确的角色和职责, 例如: 众议院议长是主持人, 负责保证各个会议活动的有序进行; 事务长则负责议院的日常运作。每次会议的模式都是预先确定好的, 允许讨论的事项都会提前告知所有参会者。对于如何对问题进行辩论也有严格的规则, 包括正反双方的发言顺序、允许发言的时间, 甚至对发言的内容也有一定程度上的限制。商榷通过投票的形式进行, 投票也同样受到限制, 如: 什么时候需要唱名表决, 什么时候可以弃权, 以及各种不同情况下赞成票需达到的比例等。

- 决策与计划

Rittel 和 Webber^[32]讨论了与“决策活动和计划活动中所固有的难以控制的特性”相关的问题, 并称之为“令人头疼的问题”。这类问题主要表现在其不确定性上: 在多数情况下, 无法区分对问题的理解和解决方案; 对于问题的表述有可能等同于问题的解决方案^[31]。大体上看, 许多方案都是可行的, 难以区分优劣。有时, 这些“令人头疼的问题”是由多方利益的冲突产生的, 例如: 将有限的房间分给不同的人, 而他们又恰巧具有相同的偏好。这类问题一般通过折中的方式解决, 即最终采用多方均可接受的方案而非最优方案。

Rittel 和 Kunz^[31]提出采用辩论的方式解决这类难题, 即不断地提出疑问, 然后探讨诸多方案的优缺点。这种方法称为基于论点的信息系统(Issue-based Information System, IBIS), 以论点、立场、论据的记录和关联为基础。以独立的论点作为树(可能为空)的根节点, 论点的孩子是立场, 立场的孩子是论据^[10]。这三个基本元素间的关联都会被标注, 例如: 论点和立场通过回应(responds-to)连接, 而论据又通过支持(supports-to)或反对(objects-to)连接到立场。

IBIS 讨论需要首先抽取出一个或多个(抽象的)论点, 参与者对这些论点提出自己的立场和论据, 并将其细化为具体的子论点。对立的立场通过达成共识或投票的方式来解决。最终的结果将是由诸多元素连接而成的森林, 它展示了讨论的演化过程、考虑过的可选方案和决策的依据。

以上例子中涉及的交互, 仅代表了一个大型交互过程在时间和空间上的一个小片段。NASA 极限协作团队中的成员已经一起共事多年, 对于彼此的特点和专长都很清楚。他们对其所在领域有着共同的工程知识、解决问题的方法和策略, 这些也是他们所在领域文化传

承的一部分。因此,“作战室”交互模式之所以能够成功,取决于该团队有着多年的交流积累。同样,对于国会会议模式,这些正式会议仅仅代表了政治冰山的一角。来自不同利益相关人的政治和经济压力,促成了会前复杂的幕后谈判,而幕后谈判又决定了会议的进展。基于 IBIS 的决策与计划活动由以下两个因素所引导:参与者基于个人经验对所讨论问题的深入见解,以及作为过程一部分所征求的专家意见和支持性文档。所有这些交互上下文,又作为团体、政府、国家等组织的一部分,逐个嵌入到更大的社会环境中。

虽然,这里的例子都是来自具体工作,但是,它们所表现的复杂性和多样性显然可以扩展到这些具体工作之外的人与人的交互中。

3.2.2 模式探索

由于很难了解到交互的完整细节和复杂的决定因素,我们往往无法获取交互中隐含的过程,不过,我们仍然可以观察到某些规律和群体的行为模式。交互通常按照某种模式进行,而不是完全不受约束的,参与者通常都会在新的场合中重复使用到这些模式^[26,27]。这些模式源自共同的信念和价值体系,往往是参与者在以往类似场合的经验积累。这些模式加强了交互行为,有助于促进后续的交互活动。参与者在交互中,不是简单地重复模式,而是显式或隐式地对自己的行为作出申明。参与者通过“可观察和可汇报”的方式进行自我监督和反思^[26],从而使用自己的行为“有理有据”^[23]。所谓“有理有据”就是明确地依照社会群体所共有的价值观和规则行动(即便是无意识地),同时,这些行动又会体现和强化已有的价值观和规则^[36]。

参与者在交互过程中要受到组织的规则、目标和规范的约束。参与者之所以能够理解其他人的行为(同时相应地调整自己的行为),是因为在某种组织环境下个人的行为都有特定的含义并被组织所认可。Bittner^[4]认为:我们对于周围事物的理解,取决于我们将它们划分到哪些规则的影响范围内(文献[15]对此做了引用)。参与者在交互过程中扮演不同的角色,如教师、学生、会议主席、参会议者等,因而还受到相应角色行为规范的约束。无论人的个性如何,在组织环境中所扮演的角色在一定程度上决定了他的行为^[2,3]。有些角色的行为可能是不合常规的,有时甚至是病态的,如影片《魔鬼代言人(the Devil's Advocate)》和《替罪的羔羊(Scapegoat)》中就有这样的例子。

语言交互也可以被视为一种博弈,每一次发言^[39]都会约束和引导后续的言谈。直观地看,问题的提出必然会引发问题的回应,包括间接回应。Searle^[39]和其他学者从语言/行为的角度(如文献[21,46,13]),定义了一个构成交谈行为本质联通区的话中行为集。独立的行为构成了可接受的“行为序列”,如参与者的某个请求可能会被接受、拒绝、或反请求(因对请求不太满意而提出的新请求)。接着,这几种情况又都可能后续动作,如“反请求”可能会被接受,原来的请求可能被撤销,或又有新的“反请求”被提出来^[47]。集体会谈由此显出某种模式,它可被视为一个演化过程。这意味着,即使看似毫无结构的交流行为,也是由语言、社会以及文化规范所引导的,由此在很大程度上规定了实际交互行为的进行方式。换句话说,大量交互构成了社会过程。社会过程在不同层次上发生,并以递归的形式互相嵌套。例如,辩论和投票可以视为会议的子过程,而会议又是组织内一个更大的过程的一部分,这个大的过程又嵌入到一个更大的组织和社会环境中。

本节着重讲述人与人(P2P)的交互,这类交互的人员参与度很高,这对于交互技术的发展也提出了专门的挑战。

3.2.3 规范化的交互

成熟的交互形成了常规,“如果社会行为在时间和空间上足够稳定,就形成了常规,即参与者习惯性地按此行事。常规构成了习惯,即日常社会生活中大量活动理所当然的特征”^[26,第376页]。常规所带来的稳定性,使得有些准则不必再对参与者进行阐明或强调^[30]。了解交互的规范和期望结果,可以减少组织间或上下级间不必要的沟通^[22]。稳定性也为提高效率带来了可能:参与者可以专心于自己工作,而不必陷入诸多不必要的讨论和谈判中。交互的成熟提供了反思的机会,对角色的理解会更加清晰,行为的稳定有助于更好地洞察行为的本质。这种反思可能会促成:对角色的定义,如给出“会议主席”和“助理”这样与明确责任相关的专用称谓;以及对规范交互行为的正式规则的定义。

具体的角色和规范描述了个体交互行为的可接受形式,如:指定地址的书写格式,或者使用标准的文档模板(例如:要求以书面形式提交申请)。这些规则也可能会根据不同的场景来限定每个角色行为的可接受形式。例如:某些特定的交互场景中,可能会要求某些活动遵照一定顺序进行。一个交互可能隐含多个复杂的模式,其中包含了多个结构化的、递归嵌套的子交互。上文提到的极限协作模式正是如此。在极限协作模式中,子交互开始于参与者对于问题状态以及问题各部分之间依赖关系的理解和直觉。类似地,基于 IBIS 模式的交互也是由嵌套的子交互组成的,这些子交互通过扩展讨论树的分支对问题进行探讨。在国会会议模式中,交互则被严格地编排与执行,参与者使用他们的(有限)权利提出行为要求(如申请投票或反对提案)。

需要注意的是:虽然交互存在着正式的规范和具体的规则,但这并不意味着交互都是机械僵硬的。事实上,规范化忽略了许多细节,规范只是将纷繁复杂的交互行为与其所处的结构化场景分离,使场景更加清晰明了。以上文提到的规范化的国会会议模式为例,它并不局限于表面的轮流发言和代表们刻板的行为,这些只是复杂博弈的进行机制。这些行为在是由诸多交织在一起的利益相关者所推动的,涉及社会、政治、经济和个人利益。

3.2.4 协同技术的内涵

接下来,我们对协同工作过程中显式表现出来的协同技术进行讨论。我们暂时先将这种技术非正式地定义为“过程感知”技术。利用过程感知技术可以获取过程信息,并将信息以多种方式进行解释、表达和运用,由此来辅助对过程活动的控制、协调和执行。该技术涉及的范围很广,从被动的信息仓库到全自动的过程控制和执行系统都是该技术涉及的范围。3.3 节将给出过程和过程感知更为严格的定义。

过程感知的协同技术可以对工作进行描述,对工作的抽象表示进行操作。这种技术的优势及其风险都来自对工作机制的显式描述。过程感知技术的基本特性就在于它提供了工作的可见性。

在文献[15]中,Dourish 将过程感知技术与其解释和协调工作的机制联系到了一起。

在解释工作时,需要将构成工作的大量活动与这些活动使用的抽象活动定义关联起来。这正好符合组织中所要求的可理解准则,以及协同工作中固有的责任机制。活动(如计划活动、控制活动)依赖于准则(如可理解准则和责任机制),同时,活动的可见性又可以反过来增强准则。使用过程感知技术描述工作,可以发现个体行为中的合理模式,使个体行为以有意义的方式可见。在这个意义上,过程感知技术是“具有可理解性和责任性的技术”,为实际的推理和个体行为制定了合理的顺序^[42]。当使用推理和行为来描述不同时期的工作时,可以采用不同的形式,以实现不同的功能:

- 以往工作的可见性

由于我们对以往工作的完成情况进行了描述,通过检查这些描述可以达到审计目的。这些审计可能包括对质量管理的考量、通过追溯审计来判断某些结果是如何获得的、或者通过追溯审计来保证结果和组织政策的一致性。对以往工作的描述不仅仅局限于组织内部,跨组织的描述同样适用,例如:对由多个组织共有的油井钻台工作情况的描述,有助于计算财务补偿^[6];印刷厂向其客户介绍印刷外包业务^[5,15]。这里提到的参考文献主要讨论的是 workflow 管理系统,但是这些论据对过程感知系统和技术同样适用。

- 当前工作的可见性

当前工作的可见性与协同技术固有的协调功能息息相关,有利于工作的定位和引导。这里所提到的“协同技术”也是过程感知技术的关注点之一。对当前活动(Activity)的清楚认识,可以使参与者明确目标,采取相应的行动(Action)。当前工作(Work)状态的可见性还有助于对工作的全局把握^[34]。工作人员可以通过检查对工作状态的描述,更有把握地发现潜在问题(如在最后期限前规避风险)。

- 未来工作的可见性

过程感知技术可以储存日志和历史记录,并可以运用预测模型对历史记录进行分析。这样,就可以使用过去工作日志来预测将来的工作。对重复情境下的结果进行预测是人类活动的重要方面^[1]。反思和计划是人类所特有的,同常规的自然发生机制息息相关。过程感知技术为新型“合成”过程的产生和实验提供了机会。通过反思能够获得全新的过程,这些过程创新是无法在长期墨守成规的执行中得到。以群体决策支持系统(Group Decision Support Systems, GDSS)为例,通过认真细致的过程活动编排,可以引导(和限制)参与者执行特定的活动。例如,先匿名提出想法,再对想法进行分类,最后是投票表决。这样就可以避免陷入会议的普遍误区,如个人主导、偏离主题以及信息超载^[29]等。

3.2.5 问题和误区

由于对工作进行了明确的描述,过程感知技术会给实际工作的灵活性带来负面影响。过程感知技术的批判者(如 Suchman^[41,43,40]和 Robinson^[35])指出:原则上,工作的抽象描述与工作本身的灵活性是相矛盾的,变化才是不变的规律。作为过程感知技术的重要因素,对工作的抽象描述并不能满足这种灵活性,也无法描述工作中错综复杂的社会和文化因素。

批评者们认为:行为应该根据环境上下文而定,而且通常需要临场应变。“遵循办公程序并不是办公的主要目的,相反,办公程序只是达到办公目的的一种途径。办公程序的平稳流转是从业人员自我引导的结果,而并不是工作本身”^[41]。因而,计划(Plan)应该作为反思

工作的依据,或是在工作完成之后作为工作的解释,不该成为实际工作开展的障碍,或者至少不是工作执行的必要资源。“合作、协同和协调有助于实现系统目标,但是使用流程型工作流程模型的系统提供的支持甚少”^[37]。

事实证明(如文献[25]、[24]、[5]和[38]),某些情况下,过程感知技术可能会为工作的灵活执行带来障碍。在实际中,技术障碍会导致工作围绕系统拐弯抹角地执行^[25,24],例如:使用一些拙劣的补救办法(如文献[5]中所描述的,参与者为了同时负责两项任务而使用一个虚假的身份),或重新解释技术对象的性质(如文献[6]中将时间表上的零小时解释为未完成)。这些机制对于实际工作的执行是有必要的,它假设工作区对于过程感知技术是不可见的,从而违背了过程感知技术的初衷。由于不可见性导致过程感知技术无法描述部分工作,有碍于过程感知技术的解释功能,相应的协调功能也就跛足难行。

协同技术研究的一个重要分支跳出了工作的相关过程,关注于提供协同工作的工具,这些工具间的协同需要由用户自己管理。我们称这些工具为“非过程感知技术”。因此,从技术的角度来看,这类系统中的过程是隐式的。这种方式确实很灵活,可用于非预期的应用^[34],因为这类技术对于不同特性的工作境况具有更强的适应性,所以被应用于许多场合。显然,这种“忽略过程”方法的不足之处在于:用户需要自己担当构造交互过程的重任。特别是如果所提供的机制与群体所期望的模式不相符时,就更加困难了。例如,当一个简单的重复活动需要繁琐的工具映射来支持时,就有可能出现上面的问题。对“非过程感知技术”的其他批判主要针对工作描述所扮演的潜在策略角色^[44]。工作描述是不全面的,会突出(或隐藏)工作的某些方面。有一种批判如是说:如果将这些带有策略偏向性的工作描述结合到技术中,只有它所服务的特定关注点能得到保证,没有从干系人的角度考虑问题,从而可能导致与实际工作的不一致。这里的主要问题可能是由于干系人本身没有直接参与到工作当中,因而在很多时候对于工作的复杂性没有足够的了解。除了工作参与者,其他人是看不到工作的具体执行情况的,特别是当工作进展顺利时。很多在局外人看起来是不费吹灰之力的工作,实际上可能是极其复杂而精细的工作(如 Suchman^[44]提到的文档整理工作)。

对工作细节缺乏正确的理解会导致对工作的描述与实际不一致,这种问题往往还会因技术开发团队与用户的脱节而加剧^[30,44]。由此会导致这样的风险:技术只是工作的一种具体表达,没有从任何干系人的角度考虑问题,进而可能导致负面的结果。嵌入到更大更复杂系统中的自动或非自动的“技术支持的过程”,会因与所嵌入系统的不一致而成为系统崩溃的潜在祸根。直观地看,当过程与它所处的环境差异过大时必定会导致一些问题。

3.3 人与人系统的特点

3.3.1 CSCW

CSCW(计算机支持的协同工作)领域关注于能够协助群体完成工作任务的理论和技术,它将社会科学与技术科学有效地结合在一起。该领域的相关期刊会议^[12]指出了用来辅助协调群体活动的计算机工具的过剩情况。这些工具从 E-mail 系统到实时群组编辑器(如 CoWord)都有。我们将这些计算机工具称为“人与人系统”(Person-to-Person 系统,或 P2P

系统)。这些 P2P 系统中,只有个别系统清楚地描述了底层的群体协同过程。本节将提出用于描述不同 P2P 系统的概念和相关术语,着重讲述过程感知的 P2P 系统,它们清晰地描述了内嵌的群体过程。

3.3.2 定义

过程(Process)是为达到某个目标而进行的活动(Activity)的集合。这些活动按偏序方式组织,即要求某些活动作为其他活动的前驱。每个活动可能包括多个属性(如输入数据、任务执行者、时间约束等)。活动可以分解为更低层的子活动或子过程,可能包含从全人工到全自动的各种执行方式。决策活动和分支结点也包括在其中,但本文不予讨论。

过程感知系统是清晰描述了其内嵌的群体过程的计算机系统。

3.3.3 例子

为了更好地说明过程及过程感知的概念,我们将讨论两个虚构的 P2P 系统。对于每个用例,我们仅描述过程中的活动的线性序列,省略并行活动和决策活动。

第一个用例是一个软件发布控制系统(Software Release Control System, SRCS),某大型软件公司每六个月使用该系统发布一次其新版软件产品。它不仅是一个存储代码、文档和消息的信息库,还是一个可自动发起活动、运行编译器及发布新版软件的过程控制系统。它是一个可以全天候访问的异步系统。该系统包含以下活动:

活动 1 是错误报告活动,持续两个月。在此期间,任何人都可以通过系统提交软件的错误报告,并进行报告评估。

活动 2 是错误修复活动,持续两个月。期间程序员修复软件产品中的错误,所有的修复需要在截止日期前提交。

活动 3 是预发布编译和系统集成测试,持续一个月。这个活动还会对测试期间未发现的软件错误进行修补。

活动 4 是最终发布编译和相关文档,这是第六个月的主要工作。

活动 5 是软件分发,在第六个月的最后几天,软件会以电子文件的方式分发给客户。

虽然这个过程控制管理系统(Process Control and Management System, SRCS)对发布过程中的五个任务全部实现了自动化的控制,但它仍然被归类为 P2P 系统,这是因为这个工作中的每个活动都是人力密集型的,需要很多高级技能人员的协作。这五个活动都是非常高层次的活动:不难想象每一个活动都包含有子过程,而这子过程又由一系列细分的活动组成。

第二个用例系统是一个虚构的会议系统,用于采用虚拟会议法进行有组织的头脑风暴并解决问题^[14]。这种实时分布技术,使得团队中的成员可以在家中通过工作站、语音通信线路和网络在互联网上召开决策会议。

这类系统称为决策支持系统(DSS),已在技术评估会议中得到成功应用^[16]。它的工作方式如下:首先,关于某个主题的一组专家被聘请或自愿参与到技术评估会议中;其次,所有参与者按照约定时间通过计算机连接到会议系统上;之后,由经过专门培训的会议主持人

负责协调控制会议的进程,保证其顺利进行,他还控制着整个会议期间所需工具的分配和使用。在虚拟会议中,主持人带领着团队开展以下活动:

活动 1 是问题探讨,即由主研究员通过“展示工具”对要探讨的问题进行介绍和说明。

活动 2 是通过头脑风暴产生解决方案,鼓励所有的参与者提供方案,其他人可以要求进一步阐明方案。这个活动的目的在于获得尽可能多的创新解决方案,不允许对任何提议进行批评。会议系统提供有头脑风暴工具,允许所有参与者即时输入解决方案或阐明请求。系统会保证所有组成员能够方便地获得这些信息。

活动 3 是方案集中,该环节将使用讨论工具对所有的方案进行讨论、辩论、分类并探讨其可行性。

活动 4 是投票,参与者对经上一环节讨论后剩下的方案进行投票,选择最可行的。这是一个复杂的由计算机控制的子过程。系统为此提供了投票工具。

系统提供了如下工具:

- 活动 1 用到的展示工具
- 活动 2 用到的头脑风暴工具
- 活动 3 使用的讨论工具
- 活动 4 使用的投票工具
- 语音会议工具

主持人决定每个活动的开始和结束时间,控制所有的工具,可以在任意时间启动或停止所有用户工作站的任何工具,并会为特定的活动启用相应的工具。除会议发言工具外,其他所有的工具都支持非口头的文字输入交流方式,这些文字信息便形成了正式的会议记录。主持人可以在适当的时候允许个人通过语音工具面向整个群组发言,这便形成了非正式的群体沟通渠道。

会议具有人与人交互密集型的特点,会议系统是 P2P 系统的典型样例。绝大部分会议系统是非过程感知的。而本例中的会议系统是过程感知的,系统中有活动的概念,并可以通过工具的使用情况了解哪个活动正在进行。

3.3.4 P2P 系统特性

有了以上两个系统实例,接下来我们定义一些术语来描述和刻画 P2P 系统。上述 P2P 系统的一个重要特性就是过程感知。一个 P2P 系统能够成为“过程感知的”,当且仅当它对支持的内嵌过程有清晰的描述。在第一个例子中,SRCS 系统显然是过程感知的,因为它会在不同活动切换的同时改变访问权限或向个人发送通知等。这五个活动显式地嵌入在软件控制系统的代码中。在其他系统中,过程被描述为一个表格、一组规则或一个显式的内部存储模型(如 workflow 系统)。

本书其他章节所讨论到的 workflow 管理系统,都是典型的过程感知系统。其中某些系统是成功且卓有成效的。在本书中,我们将 workflow 管理系统归结为 P2A 系统(可能存在争议)。

另一方面,某些 P2P 系统,如典型的实时群组编辑器^[11],没有过程的概念,更别说对过程的描述了。任何过程(如操作流转、章节分工或编辑优先级)都只是群体默认的非正式的

过程,系统对此一无所知。我们将这些系统称之为非过程感知的。

对于某些 P2P 系统,过程所包含的活动虽然是可见的,但这些活动并不用来控制系统。在会议系统的例子中,不同的工具对应于不同的活动,因此系统具备了过程感知能力。此时,工具的调用顺序和使用时间都有记录,但是并没有用于系统控制。如果对过程进行细化,在包括了读文本、听声音这类的活动后,这个会议系统就是部分过程感知的了。此时,系统可以监测到讲话或输入的活动,但并不能监测到阅读和倾听这样的活动。此时,如果主持人使用发言工具插入了其他活动,那么系统对此便无从知晓。总之,由于各种各样的原因,系统可能只感知到过程的某些而非全部的活动。我们称这些系统为部分过程感知的。

P2P 系统的另一个重要特性是执行优先级。如果指定活动 A 在活动 B 之前执行,过程感知的 P2P 系统可能会强制执行这个优先级约束;或者,系统也可能会强调灵活性,允许用户进行执行优先级的设置;还有一些混合型的系统,部分动作(Action)由系统执行的,其余的则交由用户处理。在上面的例子中,软件发布控制系统是一个具有执行优先级的系统,而会议管理系统则不然。

活动开始和活动结束这两个概念都是与强制相关的。对于每个系统,我们可以观察到活动是由系统强制终止(和开始)的,还是由用户控制的。关于强制执行,有三种可能性:系统控制、用户控制或混合控制。软件发布控制系统是系统控制的,会议系统则是用户控制的。

人类是非常善于处理异常和解决问题的,办公的大量工时和资金都用在了异常处理上。相对于 A2A 系统和 P2A 系统,P2P 系统对于灵活性的要求更高,以便处理异常和促进问题的解决。因此,当 P2P 系统中的执行优先级、活动开始和活动终止都由系统控制时,就需要一个合理的、考虑周详的系统操控(Override)机制。在有些系统中,这只需要按一下键即可,而在另外一些系统中则需要重新编码。

组织是不断变化的,因而对 P2P 系统的另一特殊需求就是动态变更。动态变更有多种形式,可能会非常复杂。我们参照变更时机对多个系统的灵活性进行了研究,发现不同的系统允许用户在不同的时机进行过程设计或过程变更,如设计时、实例化时或执行时。前述的软件发布控制系统在设计阶段对活动及其持续时间进行了编码定义,我们将这种隐含在工具中的过程称作“内嵌过程”,这类过程无法提供灵活的动态变更。不过,当某个具体的版本发布周期开始时(称为实例化时),项目经理员可以设置不同活动的确切终止时间,据此,这种系统是允许实例化时进行动态变更的。

关于变更时机,我们特别地对于以下几种时机进行考查:在过程中创建新活动的时机、指定或变更活动间顺序的时机,以及定义或变更活动持续时间的时机。活动也包含许多其他属性(如涉及的员工、信息输入等),因而我们也会对这些属性能够进行更改的时机予以考虑。

主持会议是一门高深的艺术,成功的会议时常需要进行议程的动态变更。在会议系统中,主持人可以控制代表活动的各种工具。因而,活动的持续时间以及活动的顺序都可以在执行时动态决定。这是一个非常灵活的系统,允许在设计、实例化(当会议开始时)以及执行(会议期间)时进行参数设定。主持人也可以根据需求灵活地跳过或重复一个活动。另外,主持人也可能无意识地使用工具,例如:可以在任何时候使用发言工具引入即兴活动(如

“自我介绍”或“问答环节”）。

既然 P2P 系统注重人与人之间的沟通与协作,那么在系统中提供辅助活动(Activity)执行的工具是非常重要的。因此,工具集成能力便成为 P2P 系统的重要特性之一。一个系统可能由多个不同工具组成(多工具系统),也可能仅仅是一个工具(单工具系统)。注意,单个工具也可以在多个用户屏幕同时显现(如聊天工具会在所有交流者的屏幕上同时存在)。对用户而言,软件控制系统是单工具系统,而会议系统则是多工具系统。

P2P 系统的其他特性包括活动执行方式(手工的、自动的或半自动的)和过程可见性(过去的、现在的和将来的)。3.2 节中对可见性进行了讨论。其他特性与过程关联不大,因而在本章不做讨论。

3.4 P2P 系统实例

本节将使用上文介绍的术语来刻画五个 P2P 系统实例:CoWord、WebEx Meeting Center、IPMM、LeadLine 和 Caramba。这五个系统代表了 P2P 系统的发展现状,覆盖了不同的应用领域,包括协同编著、在线会议、CASE、聊天和虚拟团队等。从复杂性和动态性来看,这五个系统具有各自不同的特性。它们来自于不同的开发团队,包括格里菲斯大学、WebEx、香港理工大学、微软以及 Caramba。它们在各自的领域都是首屈一指的主流系统。

表 3.1 对这五个系统进行了对比。对于每个系统,我们都考查了其过程感知度、执行优先级、变更时机、活动执行方式以及过程可见性。详细说明将在下面一节给出。

表 3.1 五种示例系统的比较

	CoWord	WebEx	IPMM	LeadLine	Caramba
Domain	Coauthoring	Online meeting	Project management	Chat	Virtual Team
Developer	Griffith Univ.	WebEx	Hong Kong Polytechnic Univ.	Microsoft	Caramba
Process awareness	No	No	Yes	Yes	Yes
Precedence enforcement	—	—	Hvbridp	System	User
Binding time	—	—	Instantiation time, Execution time	Instantiation time	Instantiation time, Execution time
Activity execution mode	—	—	Semiautomatic	Manual	Manual
Process visibility	—	—	Past, present	Present	Past, present