

群体会话支持系统的概念模型及应用^①

李东, 周盛

(北京大学光华管理学院, 北京 100871)

摘要:提出了一个群体会话支持系统的概念模型,并讨论了基于该模型的群件信息系统的构筑方法.群体会话支持系统的概念模型可以用于不同的组织结构和不同的工作形态.会话模式不仅是现实群体活动的抽象,同时也是构筑计算机的群体会话系统的理论依据.系统设计人员可以根据离散的会话状态的数学表达,设计信息系统中的信息交互方式,根据信息系统的软、硬件配置可以比较容易地实现具体的任务,从而对基于计算机的群体会话活动提供有效的支持.

关键词:群体会话支持系统; 信息系统开发; 概念模型

中图分类号: N94

文献标识码: A

文章编号: 1007-9807(2000)03-0044-08

0 引言

现代社会和组织中的绝大多数管理活动都是基于群体工作方式的.90年代以来的信息化社会的迅猛发展,又使得许多管理活动与信息系统紧密地结合起来.随着各级组织中 Internet、Intranet 的引进,传统的管理方式正在发生着根本的变化,在信息系统支持下的群体工作方式是现代组织中采用的一种新的工作方式,也是管理信息系统目前的一个重要的应用领域.

现在越来越多的群体合作工作都需要用信息系统来协助人进行.群体工作建筑在群体中各成员信息交互的基础上,这些信息交互依据组织结构、业务性质、企业文化等不同又有不同的内容、性质和形态,但计算机支持的群件系统可根据数据处理是集中处理或分布式处理,以及信息交换是同步还是异步来进行分类.目前虽然有大量介绍各种开发工具的资料,但却缺乏有关群体会话信息系统开发方法论的研究.在近年来国内发表的有关群件信息系统的研究文章中,一般的概念性介绍或对具体系统进行介绍的内容较多,对群件设计所必须的基础理论研究较少,而国外却有

许多论文讨论这方面的问题.

本文提出了一个抽象的群体会话支持系统的概念模型,并讨论了基于该模型的群件信息系统的构筑方法.群体会话支持系统的概念模型可以用于不同的组织结构和不同的工作形态.会话模式不仅是现实群体活动的抽象,同时也是构筑计算机的群体会话系统的理论依据.用图形化的方法,群体会话支持系统的概念模型可以形象地表现出会话进行中各元素之间的关系.这不但便于系统的用户把握当前话题的焦点和讨论的进展情况,也可以帮助他们发现讨论的不完善或不一致的部分.在信息系统设计、决策支持系统设计中,系统设计人员可以根据离散的会话状态的数学表达式来设计信息系统中的信息交互方式,用信息系统的软、硬件功能可以容易地实现具体的任务,从而对基于计算机的群体会话活动提供有效的支持.

1 群体会话过程的研究目的

在群体会话过程的设计活动中经常会遇到一些具有共性的问题:首先是如何描述一个现实的

^① 收稿日期:1999-04-13;修订日期:1999-09-17.

基金项目:国家教育部95规划资助项目(98JA063004).

作者简介:李东(1954-),男,辽宁沈阳人,博士,副教授.

群体会话活动。群体由互相依存关系的个体所组成,各个个体有自己独立的性质(如思维方式、信息传递方式和决策权限等),在群体中扮演一定的角色。同时每个个体又依赖于其它个体,相互之间有相对固定的正式关系和非正式的信息交流渠道。企业的管理活动正是在这种会话过程中实行,由会话参与者的信息交流与实际的物质流动以及特定的工作有机结合,构成特定的业务活动系统。如何用形式的方法描述现实活动中的复杂事象,就成为在设计群件系统时面临的首要问题。

其次,在手工信息处理或自然信息处理时期,会话参与者之间的信息交互是自然形成的,信息交流的方式是非固定的,有工作关系的人们有相对固定的信息交互形式。然而在计算机支持的信息交互系统中,需要对现实的群体信息交互活动方式进行再设计,使信息系统成为有效支持群体会话活动的工具。那么,为了用计算机技术对群体成员在分布式环境中以非同步方式的信息交互进行支持,应当如何设计合理、安全的信息交互途径?这种信息交互途径有没有一般的表现方式?为使设计符合群体会话工作的目标,会话参与者应当赋予哪些权力和义务?会话参与者应赋予哪些功能和限制,才能有效地支持他们?解决这些问题的基础就是将群体会话方式模型化。

另外,计算机信息系统的最大特点之一是它处理信息的效率。这与手工信息处理时因为缺乏很好的信息交互途径,可能产生职责不清、分工不明等问题相反,低效率的问题得到了解决,反之信息过滤、信息分离、信息整理等却成为难以解决的问题。如果没有很好的过滤器来进行信息削减,在金字塔型组织顶端的领导者就会淹没在信息的海洋中,因此,还涉及到如何设计度量群体活动中信息处理结果的算法问题。

2 群体会话活动模型

一个典型的群体会话活动模型可以表现为一个四元组

$$G = \langle U, R, D, S \rangle$$

其中各个元分别是参与该活动的会话参与者、会话规则、会话内容和会话活动状态空间的有限集合。

2.1 会话参与者

设 $U = \{u_k | k = 1, \dots, l\}$ 为一个会话参与者的有限集合,在该集合的元素 $u_k = (a_0, a_1, \dots, a_n)$ 中, a_1, \dots, a_n 为描述各参与者的属性, a_0 为参与者标识。一个群体会话系统中的参与者标识具有唯一性。会话参与者可能位于不同的物理地点,这一点可以在属性中表示,但在逻辑上并不影响对于他们的系统资源的分配。

2.2 会话内容

作为一个群体会话系统,所有会话内容应以统一形式来表现。将具有统一形式的会话内容存储空间称为工作空间,可以定义为: $Workspace = \{D_1, D_2, \dots, D_n\}$, 其中所有 D_i 都是数据库的有限集合。会话内容分为两大类:公共数据可以被所有会话参与者访问处理,而私有数据只能由数据所有者所处理。在现实活动中,一项会话内容一定是由某位参与者发出的,因此会话内容的数据所有者即生成该数据的用户。

D_i 有四个属性: S_i, F_i, N, O 分别是描述 D_i 的信息,包括它所在的服务器名,文件名,数据库名和所有者。元素 $d \in D_i$ 为以统一形式表示的会话内容文档,仿照关系数据库的术语称为记录。每一个记录都是有限个域 f_1, f_2, \dots, f_n 的一个有序排列。这里 f_i 称为 d 的一个域(Field)。所有 f_i 的基本结构为三元组:

$$f_i = (L, X, Y)$$

这里 L 为该域的标识, X 为域中的数据,而 Y 为作用于该域数据 X 上的特定操作。用 $Type(X)$ 来表示数据 X 的型,只有同型元素才可以同时被操作。假定在系统中有数据集合 $D = \{d_1, d_2, \dots, d_n\}$, 如果有域 f_i 具有性质 $f_i \in d_1 \cap d_2 \cap \dots \cap d_n$, 则将 f_i 称为数据 D 的共享域。在一个数据库中,不能有一个以上的同名共享域。

例1 在一个实际的应用系统中,域中类型被设定为 $Type(X) = \{Text, Date, Numeric, KeyWord, RTF\}$, 其中 $Text, Date, Numeric, RTF$ 等是常用的数据类型, $KeyWord$ 在确定关键字时使用。

操作语言用来描述对数据的特定操作。为方便起见下面使用 LotusNOTES 的 Script 语法进行描述。例如我们希望将域“author”的用户名取出作为当前文档的作者名。此操作可定义为

@Name([CN], @username). 这里 @Name, @username 是两个内嵌函数. 操作语言有四种规定的对域值的处理方式:

- 1) Edit: 系统将对数据的控制权交给用户, 域值可以由具有合法权限的用户编辑修改.
- 2) Calculate: 系统将自动进行在这里指定的运算.
- 3) Display: 自动进行运算并将计算结果在用户界面显示.
- 4) Input: 在发生创建文档事件时, 进行必要处理.

2.3 会话规则

会话规则集 R 中的元素确定了系统中所有参与者对系统资源的使用权限. 在一个群体会话系统中, 会话参与者与会话内容(它们具体地表现为数据库中的数据项)之间的关系是一个多对多的映射. 根据参与者的意向以及整体系统安全性和一致性的考虑, 必须对所有参与者以及每个会话内容的操作权限作以明确的规定. 用有限集合 $Authority$ 来记述系统中可能的操作权限, 则会话支持系统所需的规则 R 可以表示为

$$R: U \times D \rightarrow Authority$$

例2 一个简化的群体会话系统中, 会话内容以文档为单位, 其上的基础操作权限为: $Authority = \{Read, Input, Write, Create, Change,$

$No-Authority\}$. 基础操作权限组合起来可以构成更丰富的权限.

实际上, 对所有参与者和所有会话内容数据分别设定操作权限是十分低效的. 可以用对参与者的子集合设定权限的方法来提高效率. 可以对用户的集合赋予称为角色的特定名称, 系统开发者通过对角色分配对某个会话内容数据项的操作权限即可解决问题. 如果有角色的集合 $Role = \{r_1, \dots, r_m\}$, 则参与者被赋予角色可以表示为射影 $\eta: 2^U \rightarrow Role$.

例3 设 $U = \{u_1, u_2, u_3\}$ 被分割为 $U = \{u_1\} \cup \{u_2, u_3\}$, $Role = \{Chr, Prt\}$, $\eta(\{u_1\}) = Editor$, $\eta(\{u_2, u_3\}) = Reader$. 由于角色 Chr 具有编辑共享文档 d_1 和 d_2 的权限, 而 Prt 具有读取它们的权限, 所以 u_2, u_3 可以读取但不能编辑 d_1 和 d_2 (参见图1).

2.4 用户界面

在基于图形用户界面的群体会话系统中, 参与者对任何会话内容的访问都必须通过系统提供的统一形式的界面来进行. 会话界面的一种典型形式为表单 ($Form$). 一个表单上可以同时表示一个记录的若干个不同类型域中的内容, 或者是对记录中的域进行某种操作的结果, 但在一个时刻只能有一个原始数据记录显示在表单上.

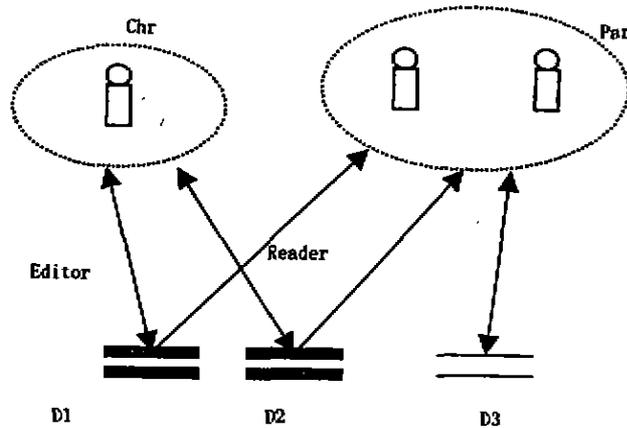


图1 用户角色和权限

表单具有操作权限属性, 操作权限由系统管理员事先分配. 因为表单是用户操作数据的界面, 用户对数据的存取必须通过表单来进行. 所以不必对所有的数据一一规定权限, 而是规定用户在

表单上的操作权限. 这样, 就明确了所有用户对数据库中的记录的操作权限.

表单上的域也可以赋予权限, 例如当某个域的数据操作权限是 $Read$ 时, 则只有身份为 $Reader$

或具有高于此权限的用户才能够使用该域。

2.5 状态空间

现实的会话活动中,人们通常是根据接收到的对方的信息,确定自己的立场或观点后再发话。仿照现实活动,会话支持系统中“接收到对方信息”或“提出意见”等,都是系统中的事件,而每一项可能的“发话”活动都可以由事件触发,再进行一项可能的活动之前,系统等待用户界面上的事件发生,此时的系统称为会话活动中的一个状态,状态的集合称为状态空间,用 S 来表示,状态空间有以下性质:

1. 有一个初始状态 Start 和一个终止状态 End.
2. 在任一个状态中,至少有一个数据记录是活动的.
3. 每个状态和一个特定的用户界面(即表单)相连.

集合 O 是系统中操作的集合,用户通过表单上发生的事件来触发特定的操作,其作用的对象是数据集中的记录,操作将引起状态空间的转换,可表现为

$$O = \{p_i : S \times U \times E \rightarrow S | i = 1, 2, \dots, n\}$$

操作对象的转换是群件应用设计中重要的组成部分,设计者必须根据实际的应用事先确定状态空间和经操作后的转换,一个数据经操作后可能激活另一个数据作为操作对象,也可能仍旧将自己作为操作对象,这种在操作后发生的数据转换可以看作是数据之间的关系。

例如,可以对集合 D 中的元素定义一个前导关系,它表示在两个状态之间存在一个能单向转换的操作,用 $Fwd(x) = y$ 表示 y 是 x 的一个前导结果,即当用户进行了 x 上的一个操作后, y 将成为当前操作的对象,如果一个前导关系上执行的操作是一个“生成新记录”的操作,而且在任何状态都可以执行该操作,则所有由该关系导出的记录构成一棵树,设 $L = \{l_1, l_2, l_3, \dots, l_k\}$ 为树上的一个路径的集合,任意一条路径代表了一个连续执行该操作所生成文档的轨迹。

对操作的定义是群件系统的核心部分,操作通过系统提供的特定语句集合来定义,由用户通过用户界面即表单或视图来触发。

以上的状态空间和操作可以用一个状态迁移

图来表示,图中的各种符号意义如下:(1)椭圆代表“初始”状态或“终止”状态。(2)矩形代表“会话”状态,在此状态下,用户可以根据视图中的信息读出必要的会话内容文档,也可以对权限范围内的自己文档进行修改。(3)箭头代表一个事件引起的状态迁移。(4)平行线代表会话内容,共享文档用粗线而私有文档用细线表示。(5)用户表示参与者或角色。(6)图案表示某类特定用户可以到达的状态,如果没有图案则表示无论什么用户都可以到达该状态。

例 4 图 2 是一个描述教师与学生会话的状态迁移图,角色“教师”的第一个操作“新问题”从“开始”状态转移到“出题”状态,角色“学生”的操作“做题”也从“开始”状态转移到“答题”状态,“纠正答题”将用在“答题”的会话内容上,当学生的“提交”操作或教师的“结束”操作发生则状态将变为“终止”。

2.6 会话兴趣指数

在群体会话活动中,如果参加人数较多,人们的意见就可能很分散,讨论无法有效地开展,这时,信息太多、太滥是造成混乱的根本原因。

对文档进行统计计算,将显示用户兴趣程度的指数与文档列表同时显示,可以帮助用户有效地进行信息概括,例如,统计一个结点的子结点,似乎可以反映出对这个会话内容的反响的热烈与否,但是加上其子树的深度和广度数字就更有说服力,还可以设计一些特定的函数来给出兴趣指数,例如,用兴趣指数函数

$$\theta_{c_i} = \frac{N(|L_i| - |c_i|)}{\sum_{i=1}^N (|L_i| - |c_i|)}$$

可描述对案例 c_i 感兴趣程度,其中用 $|c_i|$ 表示案例 c_i 的字数, $|L_i|$ 表示所有与该案例相关的话题的字数,则 θ_{c_i} 描述了一个特定案例与讨论路径中文档的平均字数的比。

3 案例研究系统的例子

案例研究是一种典型的群体活动,案例研究的主要目的是为了让参与者通过对案例的讨论,对企业管理中的一些常见问题以及产生这些问题的原因等有深刻的认识,因此这种讨论在短时间

内往往不能深入,不能指望在一次讨论中就获得一致的认识,也不能用一个简单的答案来回答案例中的问题,因此,案例研究采用非同步的群体活动方式更为合理.如果在计算机群件系统支持下,

将观点的列举、参考文献的提供、对不赞成观点的反驳以及对意见的归纳和总结等一系列过程综合起来,就可以得到更好的效果.

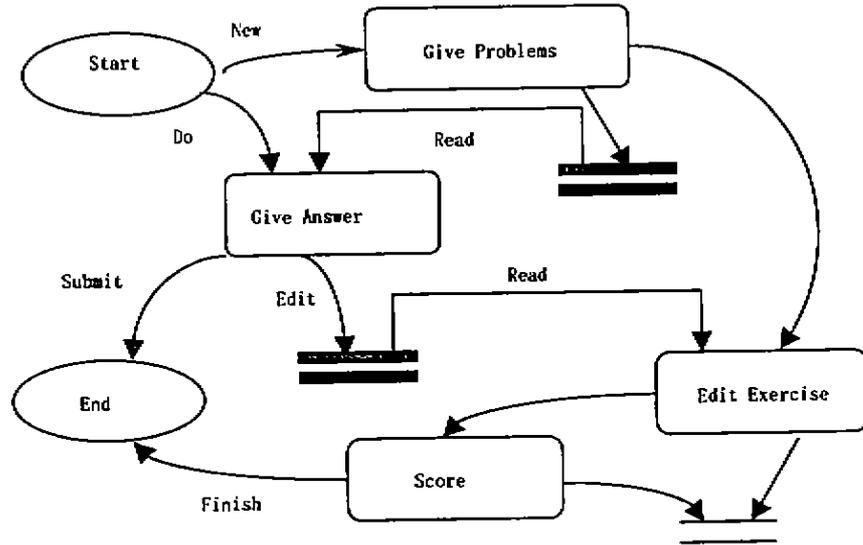


图2 状态迁移图

作为以上模型的一个应用,这里介绍一个正在实验中的群件系统 CSGS(case study support groupware system).该系统是一个用于配合 MIS 教学的系统,用 Lotus Notes 中文版开发,在 Windows NT 的局域网环境下运行.这里主要描述上述群体会话模型在实现一个实际群件系统时的应用过程.

3.1 用户集合与角色

在案例研究群体会话系统中,对用户集合作如下分割: $U = Chr \cup Prt, Chr \cap Prt = \emptyset, Chr = \{x_i | i = 1, \dots, k\} i \in I$ 和 $Prt = \{x_j | j = 1, \dots, m\} j \in J$ 分别是案例研究活动中的两个角色,即主持者(Chairman)和参与者(Participant), I, J 是自然数的下标集合.

3.2 数据库设计

在案例研究群体会话系统中,数据库 D 由两个数据子集合 C, T 构成,其中 $C \cap T = \emptyset, C \cup T = D, C$ 中的元素的定义为

$$c_i = (\text{Subject}, \text{Author}, \text{Date}, \text{Categories}, \text{Contents}, \text{Question}, \text{Reference})$$

其中各个域的意义分别是:数据库中新的案例的主题、作者、作成日期、类别、案例内容、事先准备的问题和参考资料清单.与之类似,每一个 $t_i \in T$

都有下列的结构:

$$t_i = (\text{Subject}, \text{Author}, \text{Date}, \text{Contents})$$

即话题的名称、作者、作成日期和内容.

3.3 会话规则设计

CSGS 中对上述的会话内容即案例文档和话题文档都设定了适当的权限.对于所有会话参与者来说,对已完成的文档能否存取要根据其角色来确定,这可以防止不必要的破坏.如前所述,本应用中定义了两类用户组——“Participant”和“Chairman”,对于他们赋予了不同的权限.系统中每个用户都可对自己的私有文档进行修改,但不能随便对他人的私有文档进行修改.由于主持人需要有一定的权限,所以角色为“Chairman”的用户权限为“编辑者”,可对系统中的文档进行读、写操作;而“Participant”角色的权限为“作者”,他们可以浏览文档和向数据库中加入文档,但不能修改或删除他人的文档.此外,还需对相应的表单和操作的权限加以控制,如“新案例”表单的“创建”操作对“Participant”类用户是不可见的,因此,该类用户就不能用该表单创建文档.

3.4 用户界面设计

表单设计中,属性设计决定了表单的静态性质,如表单的类型,表单的布局、表单的用户类型

等,但表单设计中实质的过程是对它的各个域的设计,由于上面已经明确地定义了会话模型两种数据类型,只要将它们与表单的域相对应即可。

表 2 表示了表单“新的案例”上的各个域的属性以及它们的作用,CSGS 的表单清单如表 1 所示:

表 1 系统中表单的例子

序号	表单名	用途	类型	可写用户	可读用户
1	新案例	主持者提出新案例	文档	Cha	All
2	新话题	参加者提出新话题	答复文档	Par	All

表 2 表单“新的案例”上的域的例子

序号	域名	作用	数据输入形式	数据类型
1	Subject	案例题目	用户输入	Text
2	Author	作者	系统	Function
3	Date	作成日期	系统	Function
4	Categories	分类	系统	Value
5	Contents	案例内容	用户输入	Text
6	Question	讨论题	用户输入	Text
7	Reference	参考文献	用户输入	Text

当表单的逻辑结构设计完毕后,就可以设计实际系统的用户界面。

3.5 状态空间和操作的定义

CSGC 中的状态空间定义为

$$S = \{ \text{Start, End, Case, Topic, Discussion, Summary} \}$$

在该状态空间中还定义了如下一些主要操作:

1. Provide_a_New_Case: $Chr \times C \rightarrow C$, 角色为 Chr 的用户 x 创建一个新案例 c_i .
2. New_Topic: $U \times C \rightarrow T$, 用户 y 创建一个关于 c_i 的新论题 t_j .
3. Discussion: $U \times T \rightarrow T$, 将个人对论题 t_i 观点连接到 t_j 上.
4. Summry: $Chr \times C \times T \rightarrow T$ 对 c_i 有关的论题创建总结.
5. Finish: $S \rightarrow S$, 结束对当前的记录操作.

操作引起的状态转换过程如图 3 所示:

以上状态变换是这样进行的(参照图 3):在 Start 状态, C 中或是完全没有内容,讨论尚未进行;或是已有过去的内容,可以接过去的案例或话题进行讨论,执行了 Case_Study 后进入 Case 状态,在此状态可提出新案例,或是对以往讨论内容

进行总结, C_i 编辑完毕后, C 增加一项新内容,因此可以开展新的讨论,当参与者读了新案例之后,执行 New_Topic 可以进入 Topic 状态,在此用户提出自己的观点或话题,可能有许多新话题会提出,随着 T 的增加,整体会话的思路进一步拓宽,但如果话题过多,就抓不住重点,因此,需要设计适当的方法限制话题的泛滥,而对于讨论的深入则需要通过 Disc_Topic 进入一个 Discussion 状态,参与讨论者对于一个话题可能有多种态度:或赞成、或反对、或补充、或质疑,这些都可以在此状态进行,这时,某些话题受到普遍重视,重点逐渐集中,随着讨论者意见的综合,新意见减少,对问题的答案逐渐形成,主持者在适当时机可以进行总结。

3.6 系统运行流程和意见的归纳

系统运行的流程是:主持人在其中输入案例说明后,讨论者就可以在自己的工作站上针对特定的案例发表自己的观点,或者对别人的观点发表意见,进行自由的讨论,系统会对所有的文档按照一定的准则进行分类显示,方便数据的查找,并可以根据用户自己的设置在某些与用户有兴趣的主题有关的文档产生时自动提醒用户,在一个案例的讨论结束后,主持人可以生成此案例所有观

点的列表,并可加入主持人的归纳意见,最后对过时的案例文档进行归档,以整理数据库。

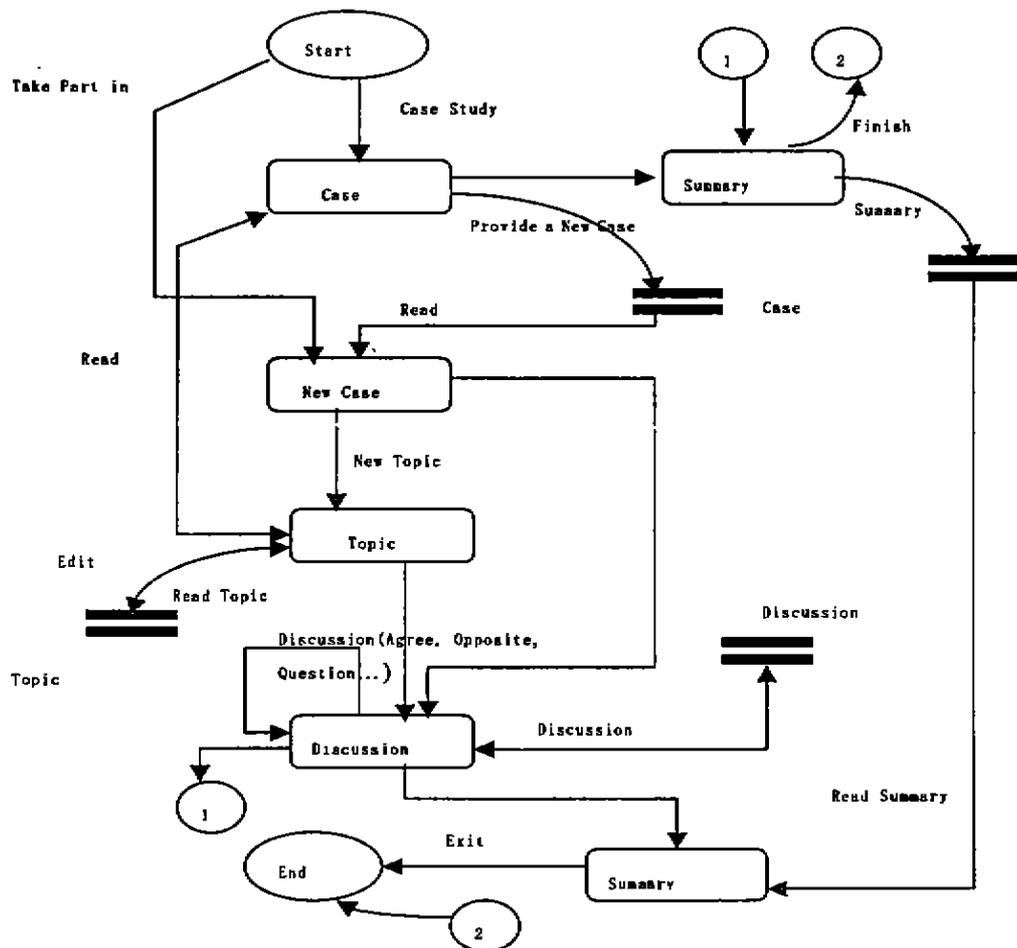


图3 案例研究系统的状态迁移图

考虑到不同用户的需要,在系统设计中需要设计若干视图,来描述不同的数据。CSGS中设计了“所有文档”,“按案例分类”,“按作者分类”,“按关键字分类”以及“归档”等五个视图。“所有文档”中用锯齿状的折叠方式尽量减少文档的每个视图,使所有文档的显示清晰明了。“按案例分类”中将一个案例有关的文档全部列在其下方,以增强其可读性。其它视图的思想与之类似。

信息过滤对于用户来说是十分重要的。信息过滤能有效地减少信息工作中的时间和工作的浪费,有助于去掉信息中的不确定因素。为进行信息过滤,CSGS自动要求话题文档作者必须指定1至3个主题词。系统根据主题词,将话题文档分割成若干个子集合。用户亦可以根据主题词来进行对特定内容的查找。

CSGS中还具有辅助总结功能。在全局视图中有一个“案例总结”操作,主持人点击操作条或从菜单上激活一个代理,即可由系统生成一份所选案例的总结,系统将自动列出目前对此案例的所有话题,及话题的提出人,提出时间等信息,并包括对相应文档的链接(可通过点击图标打开对应文档),主持人还可在总结中填入自己的总结内容。

4 结论

群体会话支持系统的概念模型是对特定群体活动的一个抽象,它明确地表现了在这种群体活动中的系统结构,有助于帮助系统分析员对于现实系统的分析和描述,或是对使用群件系统后对

现实系统的再设计。

本文中提出的概念模型还处于研究的起步阶段,我们还面临许多问题。例如,对于异步处理的时间逻辑如何表达,对于数值处理和关系数据库记录的表达方式,以及和 Internet 的结合等。

应当看到,会话模型的应用前景是十分广阔

的。不仅可用 Notes,也可以用 Java, CGI 等基于 Internet 和 WWW 技术的信息工具来实现,如果用超文本技术,则对于用户会更为友好,能够更为生动、有效地利用来实现类似案例研究之类的群体会话活动。

参考文献:

- [1] Edwards W K. Policies and roles in collaborative applications[C]. Proc. of Computer Supported Cooperative Work, 1996. 11-20
- [2] Rodden T. Populating the application: a model of awareness for cooperative applications[C]. Proc. of Computer Supported Cooperative Work, 1996. 87-95
- [3] Kouzes R T, Myers J D, Wulf W A. Collaboratories: doing science on the Internet[J]. IEEE Computer, 1997. 29-34
- [4] Schatz B R. Building an electronic community system[J]. Journal of Management Information Systems, 1992, 30-35
- [5] Yasuyuki, SUMI. Personalizing information in a conversation support environment for facilitating collaborative concept formation and information sharing[J]. The Transactions of the Institute of Electronics, Information and Communication Engineers, 1997. 85-91
- [6] Anderson J R. The architecture of cognition[M]. Harvard University Press, 1983
- [7] Baecker R M. Readings in groupware and computer-supported cooperative work: assisting human collaboration[M]. San Francisco: Morgan Kaufmann Publishers, 1993
- [8] Masuch M, Wargelien M. Artificial intelligence in organization and management theory[M]. Elsevier Science Publishers B. V., 1992
- [9] 矢野博之等. 创造性的对话收集与分析[C]. 日本京都大学社会情报学研讨会论文集, 京都大学社会情报学科, 1997. 1-4
- [10] 李东. 管理信息系统的概念与应用[M]. 北京:北京大学出版社, 1998
- [11] 成思危. 虚拟经济与金融危机[J]. 管理科学学报, 1999. 2(1): 1-6

A group conversation support model and its application

LI Dong, ZHOU Sheng

Guang Hua School of Management, Peking University, Beijing 100871, China

Abstract: In this paper, we propose a conceptual framework of collaborative conversation activities and discuss its application in an implementation of a case study groupware system. The model is an abstraction of real conversation activity and is the basis for implementation of the computer based groupware information systems. Though this research is just in its infancy, we find that it is possible to be used for describing relatively complex group activities—the behavior and the relationships of the members in a conversation. With the guidance of the framework, a system engineer can represent the structure of the conversation and can implement the real system in short period using powerful system developing tool.

Key words: group conversation support system; system development; conceptual model