微机网上的分布式专家联合系统 UNION（下）*

赵致琢　庄庆雨　曹华　陆汝钤

（中国科学院数学研究所）

A DISTRIBUTED EXPERT UNITED SYSTEM UNION ON MICROCOMPUTER NET (PART TWO)

Zhao Zhizhou, Zhuang Qingyu, Cao Hua and Lu Ruqian

(Institute of Mathematics, Academia Sinica)

ABSTRACT

Distributed experts united system is an advanced topic in knowledge engineering. In this paper, implementation methods and techniques of UNION which is a distributed experts united system on microcomputer net are described. The concept of distributed experts united system was first proposed by professor Lu Ruqian in 1985. In next year, the experimental system UNION began to be developed by Institute of Mathematics, Academia Sinica. The environment is UNOS system with C and PASCAL on MC68000 microcomputer. Three microcomputers are connected by an Ethernet. So far, the supporting system Support 1 and communication system have been finished and other units are being developed.

摘　要

本文描述微机网上的分布式专家联合系统 UNION 的实现方法和技术。UNION 是继 1985 年陆汝钤提出分布式专家系统的概念及其设计思想之后于

* 国家高技术发展计划资助课题。

次年夏天开始组织实现的一个实验系统。目标机型是配有UNOS操作系统、PASCAL和C语言的MC68000微机，它们用Ethernet局部网联网。UNION目前已完成支撑系统Support1和通信系统，其余部分正在实现中。UNION的工程实现由中国科学院数学研究所承担。

§5. UNION系统的实现

UNION系统的实现采用了模块化程序设计技术、并发程序设计技术和网络通信技术，使用PASCAL和C两种高级程序设计语言。下面详细介绍各部分的实现。

1. 多用户人机接口

(1) 结构

多用户人机接口是一个独立的模块，在UNION动态运行时，它以进程形式存在，由用户自行启动。每个用户从终端建立一个多用户人机接口进程与UNION交互，其结构可以是图3中的一种。UNION多用户的实现在人机界面一级使用了人机接口程序重入技术。

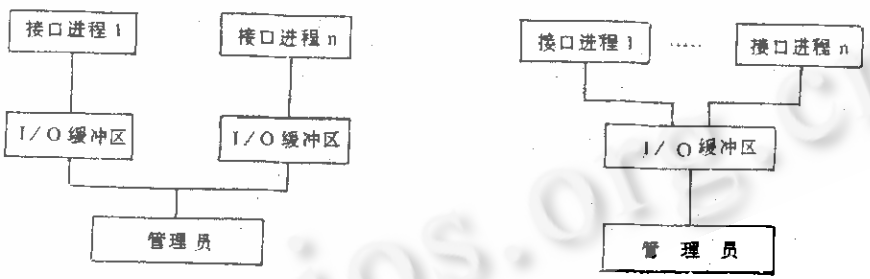

图3 多用户人机接口结构

依输入输出缓冲区的不同设置，我们有图3中(a)、(b)二种实现方法。采用(a)结构，虽然能提高效率，但每次从某个缓冲区取来的任务不一定是结点上优先级最高的任务。若提供结点上统一的优先级标准，则管理员每次得比较各缓冲区排在最前面的任务的优先级，这样就会牺牲效率。(b)结构虽然能保证优先处理优先级最高的任务，但由于多个进程竞争一个临界区(资源)，效率将会降低。UNION采用(b)结构。

(2) 实现技术

为了保证UNION的透明性，让用户操作时感到系统是一个专家系统，有必要使用一种技术，它能够模拟不同系统的人机交互。

我们采用面向对象的程序设计技术来实现人机接口。采用多层对象描述专家系统的人机接口。每层对象描述与屏幕的一帧窗口模式相对应，对象的属性说明窗口位置、尺寸、颜色、字符表示、可接受的信息和消息。每层对象有一个Method 或Procedure 与之对应。消息来自用户的操作，不论是哪个专家系统，交互模式在多用户接口中通过对象描述和Method 或Procedure 结合可实现完全统一。

每个专家系统在加盟时由管理员构造对象描述，所有对象描述按专家系统名组成一个对象名表。对来自不同专家系统的数据或信息，系统根据对象名表调用相应的Method 或Procedure 处理，实现相应的输入输出。

限于篇幅，具体描述和算法实现，我们将另文发表。

2. 管理员的实现

(1) 内部表的实现

内部表的实现采用类型文件。每一种表的结构是一个复合结构的记录类型，分别存放在不同的类型文件中，为管理员和其它子系统所共享。

(2) 管理员与专家系统的注册

管理员接受一个专家系统到UNION 中是通过处理各专家系统的一张表来实现加盟的。表的结构和内容如下:

Expert-Name: 记录待加盟的专家系统名;

Decomposing-Rule: 记录一组分解规则;

Synthesizing-Rule: 记录一组解的综合规则;

Representation: 记录知识表示方式;

Node-Addr: 记录专家所在结点号;

Cost: 专家系统的运行费;

Vocabulary-Index: 标准词汇与专用词汇对照表; 以及其它各项性能和功能属性信息。该表由管理员处理后将有关信息分别存入内部表和专用与标准词汇对照表，并存入背景知识和控制性元知识库中。在完成本结点某个专家系统登记后，管理员还要将有关信息发送到网上的其它结点，以便在那些结点上完成登记。管理员还要为每个专家系统建立专用词汇与标准词汇对照表，供专家系统在解题中使用。

(3) 知识的设计及管理

管理员的知识主要由背景知识和控制性元知识组成。背景知识包括任务的性质、分类、分解等说明领域问题的知识，控制性元知识包括对任务进行网络动态规划、控制和修改策略，描述如何利用专家系统完成特定目标的知识，等等。

管理员静态建立上述知识库，同时在UNION 运行中，特别是在新专家系统加盟过程中和完成一组问题求解后由它根据运行结果动态修改知识库。

(4) 任务管理

分布式任务分解和结果合成是管理员的核心任务。任务分解和分布受网络动态状态和用户约束条件等诸多因素的影响，往往非一次就能完成。

管理员对任务分解和分布实质上是综合多种因素对任务进行规划。在得出一组规划后，管理员要用一组约束条件来检验，以确定这个规划(包括分解和分布)能否被接受，当不能接受时，管理员将对任务重新规划。

一个被接受的规划由一组任务和执行这些任务的专家系统(包括与专家系统有关的网络信息)、工作方式、任务的实际约束时间和任务的相互关系等项内容组成。规划由合作控制程序具体组织实施。下面给出一组分解任务的形式化表示。

我们用 $P_1, P_2, \cdots, P_n$ 分别表示n个子任务，用 $P=[P_1, P_2, \cdots, P_n]$ 表示 $P_1, P_2, \cdots, P_n$ 是P的一个分解；$\{P(Mycin)\}$ 表示P的一个例示，即P由专家系统Mycin来完成；$[\{P_1(e_1)\}; \{P_2(e_2)\}]$ 表示两个例示按顺序关系组成的一个例示组，其中 $\{P_1(e_1)\}$ 在 $\{P_2(e_2)\}$ 之前；$[\{P_1(e_1)\}, \{P_2(e_2)\}]$ 表示两个例示按并发关系组成的一个例示组；显然，$[\{P_1(e_1)\}, \{P_2(e_2)\}]$ 等同于 $[\{P_2(e_2)\}, \{P_1(e_1)\}]$。于是，$\{P(e)\} = [\{\{P_1(e_1)\}, \{P_2(e_2)\}\}; \{P_3(e_3)\}; \cdots; \{P_n(e_n)\}]$ 表示P的例示等同于两个例示 $\{P_1(e_1)\}$ 和 $\{P_2(e_2)\}$ 的并发，后跟以一串顺序的例示 $\{P_3(e_3)\}; \cdots; \{P_n(e_n)\}$。至于工作方式，主要是说明子任务执行时的相互关系。我们分别用"="号上加"-"、"*"和"o"表示"流水线"、"冗余"和"讨论"三种工作方式。这样，

$\{P(e)\} \equiv [\{P_1(e_1)\}; \{P_2(e_2)\}; \cdots; \{P_n(e_n)\}]$ 表示P的例示等同于按流水线方式顺序执行一组例示 $\{P_1(e_1)\}, \{P_2(e_2)\}, \cdots, \{P_n(e_n)\}$，其中 $\{P_i(e_i)\}$ 的输出为 $\{P_{i+1}(e_{i+1})\}$ 的输入，$\{P_n(e_n)\}$ 的输出即为解；

$\{P(e)\} \stackrel{*}{=} [\{P_1(e_1)\}, \{P_2(e_2)\}, \cdots, \{P_n(e_n)\}]$ 表示P的例示等同于按冗余方式并发执行一组例示 $\{P_1(e_1)\}, \{P_2(e_2)\}, \cdots, \{P_n(e_n)\}$；

所谓冗余方式，是指各例示相互独立。一般地，此时有 $\forall i, j, P_i = P_j, e_i \neq e_j, 1 \leq i, j \leq n$。$\{P(e)\} \stackrel{o}{=} [\{P_1(e_1)\}, \{P_2(e_2)\}, \cdots, \{P_n(e_n)\}]$ 表示P的例示等同于按讨论工作方式并发地执行一组例示 $\{P_1(e_1)\}, \{P_2(e_2)\}, \cdots, \{P_n(e_n)\}, \{P_i(e_n)\}(i = 1, \cdots, n)$ 之间可以交换数据、知识和任务等信息，合作完成一个任务。

此外，结合三种工作方式还可以组成一个混合例示组，我们仅举一例，不再详细说明。

$$\{P(e)\} \stackrel{o}{=} [\Gamma_1, \{P_2(e_2)\}, \cdots, \Gamma_n],$$

$$\Gamma_1 \equiv [\{P_1(e_1)\}; \{P_2(e_2)\}],$$

$$\Gamma_n \stackrel{*}{=} [\{P_2(e_2)\}, \{P_{n-1}(e_{n-1})\}, \{P_n(e_n)\}];$$

结果综合主要依据任务的分解、结果综合规则、背景知识中的常识、约束条件以及网络全局可信度的一致性处理模型。因内容较多，限于篇幅，我们将另文整理发表。

(5) 控制

管理员利用一组系统调用控制其子进程。它按照系统命令启动所在结点的专家系统，建立专家系统进程，并向合作控制进程报告。系统初启时，管理员还要建立通信进程和合作控制进程，而在整个UNION系统结束工作前，撤消有关的进程。

3. 支撑系统的实现

请参见文献[19] [21]。

4. 通信系统的实现

(1) 结构

以三个结点为例，通信系统的网络结构如图4所示。

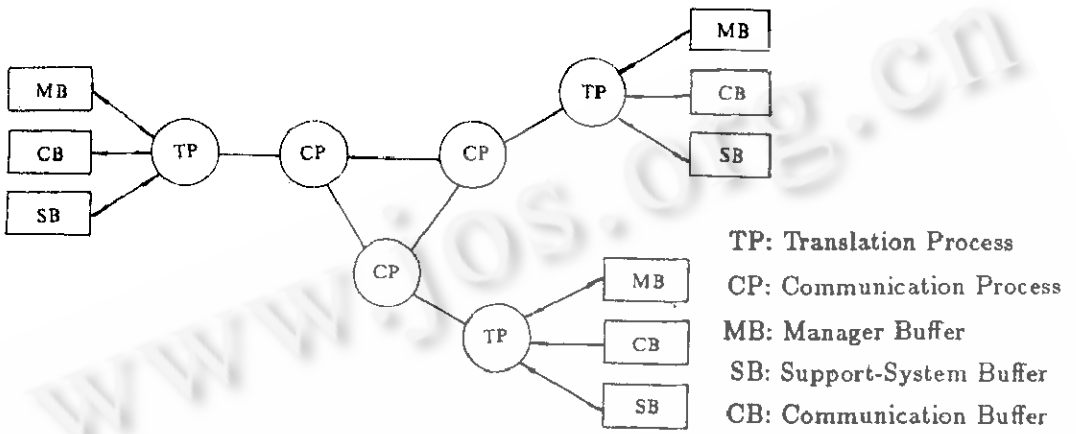

图4 通信系统结构

每个结点机上通信系统实际上是UNION初启后到结束前始终处于执行状态的一个进程，它不断地检测是否有等待发送给其它结点的消息和是否有待接收发往本结点的消息。该进程由管理员程序在系统初启时创建。其中，TP是知识表示的转换程序，CP是通信程序，CP可以设计成TP的一组过程，将TP和CP组合在一起。

(2) 实现技术

我们使用的局部网称为Universenet。Universenet中两个结点间的通信是靠一对联接伙伴(Listen /connection)来实现的。一个结点机向另一结点机发送消息前，首先打开一个信箱作为监听(listen)，另一个结点机打开该信箱作为联接(connection)。若这两个操作都成功，用户程序就可以通过这对联接伙伴通信。信箱由用户逻辑设定而由Universenet实际提供。通信结束后，应首先关闭connection，然后关闭listen。

每一对联接伙伴需要消耗一定的资源，整个网络系统所能建立的联接总数是有限的，可建立的联接数和每一对联接伙伴所占有的资源量是在系统初启时确定的，系统管理员可以利用Universenet提供的工具修改联接数和每一联接占用的资源量，然后重启系统。

为了实现可靠通信，确保通信消息不被丢失，在UNION系统中我们采取如下通信方式：

任意两个结点机之间传递消息时，每一结点机都有各自的接收信箱，如图5所示。

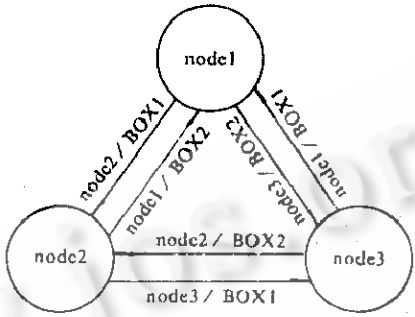

图5 结点联接方式及信箱设置

据此，任意两个结点机之间都可进行通信，不依赖其它结点机。UNION系统初启后，上述联接分别由各结点机上的通信系统在初始化中建立，这些联接伙伴一直到UNION停机前才消亡。这种方式虽然花费较多资源，但提高了通信效率，同时防止了某一消息长时间未被传送的问题。然而，当网络结点数n较大时，所需信箱数为$n(n-1)$，显然这种方式是不合适的。

下面我们以结点*node2*向*node1*发送消息为例，说明联接的过程：

5. 矛盾消解系统的实现

首先我们给出矛盾的定义。

定义. 一个事实性知识表示为一个二元组(P, Be)，这里P为模糊命题，Be为[0, 1]区间的数，它刻划P的真值的模糊性。

应当指出，这一定义实际上假定领域专家对给出的事实知识的自信度恒为1。这一简化不影响我们对辩论方法的介绍。

定义. 令P为模糊命题。称P的内涵$I(P)$为P的解释，I称为P的解释函数。I的定义域是全体模糊命题组成的命题空间，I的值域是全体模糊命题的解释组成的解释空间。

定义. 设$k_1 = (P_1, Be_1)$和$k_2 = (P_2, Be_2)$为二个事实性知识，I为解释函数。称k_1与k_2是相异的，如果$I(P_1) \neq I(P_2)$。

定义. 设$k_1 = (P_1, Be_1)$和$k_2 = (P_2, Be_2)$为二个事实性知识，I为解释函数。称k_1与k_2是不一致的，如果$I(P_1) = I(P_2)$。且$|Be_1 - Be_2| \geq \lambda, \lambda \in (0, 1)(\lambda > 0)$。

定义. 设$k_1 = (P_1, Be_1)$和$k_2 = (P_2, Be_2)$为二个事实性知识。称k_1与k_2是矛盾的，若

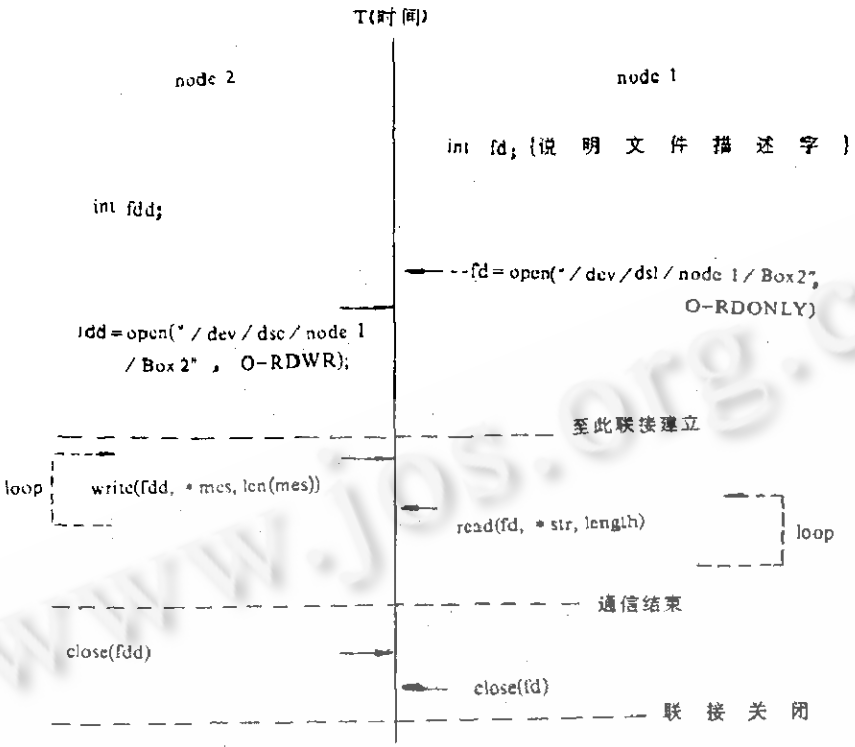

图 6　通信过程说明

(a) k_1 与 k_2 是相异的，或

(b) k_1 与 k_2 是不一致的。

λ 的选择对不同的命题是不一样的，它可定义为一个函数，也可以事先存入背景知识中。在UNION中，对任一命题P，我们只关心刻划P的真值及其模糊性，而不关心 ¬P 的真值及其模糊性。如果考虑 ¬P，或者更一般地考虑P有K值的情况，只需将[0,1]区间划分成K个区间，然后根据P的Be值落在这K个区间中的哪一个来定义K个值。

UNION系统中各专家系统求得的解用事实性知识表示。当合作控制程序检测到各专家系统求得的解之间存在矛盾时，即调用矛盾消解系统处理。

矛盾消解系统处理矛盾的基本思想是将自己作为辩论者之间的裁判，依据合作控制程序保留的推理树，沿叶子方向分别检测每一结论(中间结果)，利用假言推理、非单调推理和验证技术，判定推理过程中有没有与常识或双方共有的知识相矛盾的结论。具体判定和验证由矛盾消解系统确定其中的某些专家系统去完成。在无法判断谁是谁非时，根据表决或综合解决矛盾。限于篇幅，我们将另文发表详细方法和技术。下图是矛盾消解系统工作示意图。

6. 专家系统

UNION中加盟的专家系统，不论是按规范设计还是加盟前进行修改，应具

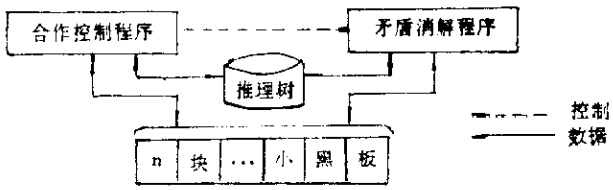

图7 矛盾消解系统工作示意图

有如下功能:

(1) 专家系统应能与分配给它的一块标准小黑板进行I/O通讯, 它能理解小黑板上各项命令的含意并作相应处理, 如求解问题和读取数据等; (2) 专家系统在求解问题时可以经小黑板向系统发出请求, 也可以通过增加的特别请求或发出特别请求, 向用户提出一些进一步的询问; (3) 专家系统在与小黑板进行I/O通讯时, 应能根据词汇索引表进行非标准词汇与标准词汇之间的转换。网络通信统一在UNION的标准词汇表下; (4) 专家系统对临界区的操作应增加P-V操作。

顺便提一下, UNION的工作将为今后专家系统设计规范提供依据。

7. 一个简单的例子

假定UNION用于银行借贷。设网络结点node1, node2和node3上, 分别有Exp1(自行车市场分析系统)、Exp2(偿还能力分析系统)和Exp3(银行资金安全分析系统)三个专家系统。为了简化, 假定银行在收到借贷申请后, 根据借贷者投资效益、偿还能力和银行资金的安全决定是否同意。现设有一自行车厂到银行申请贷款x元用于扩大自行车再生产。于是, 管理员首先根据下列知识对问题分解:

$$IF(\text{投资效益} * \text{风险系数} > 1.0)\&$$
$$(\text{资金偿还能力} > 0.85)\&$$
$$(\text{银行资金安全} > 0.9)$$
$$THEN(\text{同意借贷})$$

依此条知识, 管理员将任务分解后得到分解:

$$\{(\text{借贷审查})\} \stackrel{\circ}{=} [\{\text{投资效益评估}(Exp1)\}, \{\text{偿还能力评估}(Exp2)\},$$
$$\{\text{银行资金安全评估}(Exp3)\};$$

然后, 它将分解结果递交给支撑系统。

Support1收到管理员发来的一组任务后, 首先建立由Exp1, Exp2和Exp3组成的一个合作小组, 为Exp1, Exp2和Exp3分别分配一块小黑板, 记为bb1, bb2, bb3, 将初始数据和命令分别写入bb1, bb2, bb3。Support1中的合作控制程序在完

成黑板分配后，通知管理员启动Expi (i= 1, 2, 3)。每个Expi (i= 1, 2, 3) 由所在结点的管理员启动。

Support1 中的数据传送进程和通信系统将bb1, bb2, bb3 中的数据分别传送到Expi (i= 1, 2, 3) 各自自带的小黑板上(分别记为b_1, b_2, b_3)，同时将b_1, b_2, b_3 上专家系统输出的数据和信息传送到bb1, bb2, bb3 上，保持b1, b2, b3 与bb1, bb2, bb3 互为不同结点上的映象(不考虑时间延迟)。被启动的Expi (i= 1, 2, 3) 在读取b_i 的内容后，根据b_i 上的系统命令执行子问题的求解。设Exp1 得出投资效益为2.8, 风险为0.2, 则$2.8 \times (1 - 0.2) = 2.24$, 满足投资效益 * 风险 > 1.0 的约束。又设Exp3 得出银行资金安全系数为0.94, 满足银行资金安全 > 0.9 的条件。Exp2 是对申请者的借贷偿还能力进行评估，但由于偿还能力的判定要由申请者的经营前景，包括投资效益，往年盈亏和是否欠债等因素决定，因此，Exp2 将在b2 上提出投资效益评估问题，请求其它系统协助解决。此外，Exp2 还将发出特别请求，要求与用户交互，了解申请者以前的经济状况，获取必要的信息。b2 上的问题传到bb2 上之后，合作控制程序必将在某一时刻读到这一问题。合作控制程序读取bb2 上的问题后，首先判定该问题是否在目标(与/或树) 中出现过，如果未出现，则插入树中，然后根据背景知识和控制性元知识确定由Exp1 执行求解，并将求解命令写入bb1 中。现在恰好Exp2 提出的问题曾在目标树中出现过，于是合作控制程序就将Exp1 求解该问题的结果写入bb2 中，从而实现了合作。现设Exp2 得到Exp1 求出的结果后得出资金偿还能力为0.75 (可能由于该厂以前有欠款未还清所致), 于是，这组任务求解结束，由Support1 将结果递交给管理员。

管理员接到这组结果以后，根据分解规则进行综合，决定不予贷款，因为该厂偿还能力为0.75, 小于0.8。

§6. UNION 的进程结构与并发控制

UNION 中每个结点机上的进程结构如图8 所示。

启动UNION 即创建了管理员进程。系统初启后，管理员进程首先创建通信进程和合作控制进程。多用户人机接口进程由终端用户自行启动，它通过事先设置的文件与管理员连接。专家系统进程是由合作控制进程通过信箱向管理员发出信件，然后由管理员依信件创建或撤消的。矛盾消解进程则是在求解结果有矛盾时，由合作控制进程创建。

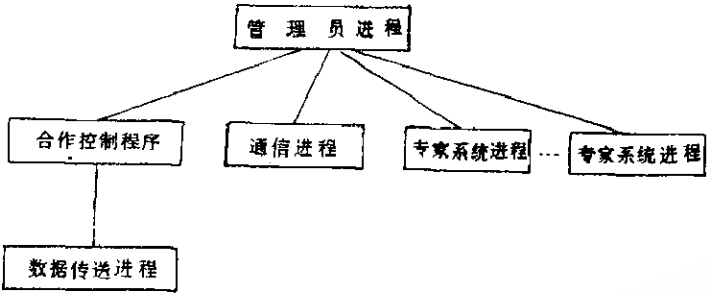

图8 UNION中结点机上的进程结构

§7. 死锁检测与预防

由于子任务的执行之间存在明显的逻辑依赖关系，以及各专家系统在执行任务时与其它专家系统之间存在的相互依赖关系，因此，当这种依赖关系在系统动态运行中存在隐式循环时，UNION就可能出现死锁。如多个专家系统在求解一组问题时因相互等待对方提供帮助而形成永久性循环等待队列。

UNION系统中采用任务受限时间法解决死锁问题。这一方法是让管理员在进行分布式任务分解时，对产生的实际任务分解和分布给出每一任务的实际时间约束条件，然后由管理员集中检测处理。这样做能有效地防止死锁的发生。

§8. 结束语

分布式专家联合系统的研究在国外归属分布式人工智能(DAI)一类。DAI领域大规模地开展研究工作已经历了十一个春秋，取得了很大的进展。

我们认为，DAI的一般原理和结构中的许多问题已经清楚，但在系统的研究与开发方面，尚存在一些待解决的问题，主要是知识表示的转换，可信度的一致性处理，矛盾的消解等，但这些基本上是AI一般常遇到的问题，并非专属DAI。设计出一个实用化的分布式专家联合系统目前看来尚有一段艰难的路程要走，原因是一个功能齐全、庞大而又完整的复杂系统是太困难了，特别是AI和知识工程已有的一些成果尚不足于支持系统的实用化和商品化。

UNION计划的意义比较重要的有以下几点：

(1) 为DDBMS和DDSS提供一个核心系统，并为DKBMS的研究给出一条新的途径；

(2) 为专家系统设计规范的提出提供实践经验和理论依据；

(3) UNION 的研究涉及到一些关键技术和理论问题，并由此提出一些新课题。

我们相信，UNION 计划将促进知识工程的深入研究。

鸣谢： 本文的研究工作得到计算机科学室许多学者的关心和帮助，李震宇和孔琢参加了UNION 总体结构的前期讨论，在此谨致衷心的谢意!

参考文献

[1] Commarata, S. et al., Strategies of Cooperation In Distributed Problem Solving, IJCAI-83, Karlsruhe, West Germany, 1983, pp. 767–770.

[2] Corkill, D. D. and Lesser, V. R., The Use of Mata-level Control For Coordination In A Distributed Problem Solving Network, IJCAI-83, Karlsruhe, West Germany, 1983, pp. 748–756.

[3] Davis, R. et al., Negotiation as a Metaphor for Distributed problem Solving, AI, vol. 20 (1), 1983, pp. 63–109.

[4] Durfee, E. H. et al., Using Partial Global Plans to Coordinate Distributed Problem Solvers, IJCAI-87, August, 1987, pp. 875–883.

[5] Durfee, E. H. et al., Increasing Coherence in a Distributed Problem Solving Network, IJCAI–85, August, 1985, pp. 1025–1030.

[6] Erman, L. D. et al., The Hearsay–II Speech Understanding System: Integration Knowledge To Resolve Uncertainty, Computing Surveys, vol. 12 (2), 1980.

[7] Erman, L. D. et al., The Design and an Example Use of Hearsay–III, IJCAI–81, August, 1981.

[8] Georgeff, M., A Theory of Action for Multi-agent Planning, AAAI–83, August, 1983, pp. 121–125.

[9] Georgeff, M., Communication and Interaction in Multi-agent Planning, AAAI–83, August, 1983, pp. 125–129.

[10] Hayes-Roth, B., The Blackboard Architecture: A General Framework For Problem Solving?, HPP Report, HPP–83–30, May, 1983.

[11] Hayes-Roth, B., A Blackboard Architecture For Control, AI, vol. 26 (3), 1985.

[12] Hayes-Roth, B., The Blackboard Architecture: A general framework for problem solving, Heuristic Programming Project Report, HPP-83-30, Stanford University, 1983.

[13] Hayes-Roth, B., BB1: An Architecture for Blackboard Systems that Control, Explain, and Learn About Their Own Behavior, Heuristic Programming Project Report HPP-84-16, Standford University, 1984.

[14] Huhns, M. N., Distributed Artificial Intelligence, Pitman Publishing, 1987.

[15] Lesser, V. R. and Corkill, D. D., Functionally Accurate, Cooperative Distributed Systems, IEEE Transactions On Systems, Man, and Cybernetics, vol. SMC-11 (1), 1981, pp. 81–96.

[16] Lu Ruqian, Expert Union: United Service of Distributed Expert Systems, Technical Report 85-3, Depertment of Computer Science, Minnesota University, June, 1985.

[17] 陆汝钤，分布式专家系统，《新一代计算机文集》，《计算机研究与发展》编辑部，1988, pp. 114–124.
[18] Lu Ruqian, The Architecture and Strategies of Expert Union, Advances in Chinese Computer Science, Singapore World Scientific Pub. Hause, 1988.
[19] 陆汝钤，赵致琢，分布式知识工程研究(I)，《知识工程进展(1988)》，中国地质大学出版社，武汉，1988, pp. 347–356.
[20] 周龙骧，微机网上的分布式关系型数据库管理系统C-POREL，中国科学，A辑，(1) 1985, pp. 955–964.
[21] 赵致琢，分布式知识工程研究，中国科学院数学研究所硕士学位论文，1988. 7.30.
[22] Luis, E. C. N., On Distributed Artificial Intelligence, Knowledge Engineering Review, 1989 (1), pp. 23–59.
[23] Bruce, S. and Peter, W., Research Directions in Object-Oriented Programming, The MIT Press, Cambridge, Massachusetts, 1987.