基于案例的动态科学工作流模型

文元桥[1,2]　余胜生[1]

（华中科技大学计算机科学与技术学院　武汉 430074）[1]　（武汉理工大学航运学院　武汉 430063）[2]

摘　要　为提高科学工作流对不确定性因素的处理能力，本文建立了一种树状结构的动态科学工作流模型，它通过与基于案例的推理技术相结合，能很好地解决科学工作流对动态性的要求，提高了科学工作流管理系统的自适应性。基于案例推理的重用，为解决科学工作流低重复性问题、实现科学工作流从单个计算步骤到整个流程定义的多层次重用提供了有效的解决手段。

关键词　科学工作流，动态，基于案例的推理

A Case-based Dynamic Scientific Workflow Model

WEN Yuan-Qiao[1,2]　YU Sheng-Sheng[1]

(Huazhong University of Technology, Wuhan 430074)[1]　(Wuhan University of Technology, Wuhan 430063)[2]

Abstract　To improve the scientific workflow for processing uncertainty, a tree-structure dynamic scientific workflow model is proposed. By combined with the case-based reasoning, the model can satisfy the requirement of dynamic scientific workflow and improve its self-adaptability. The case-based reasoning reuse provides an effective method to reuse the scientific workflow definition from separate steps to a full-process definition considering the lower repeatability of scientific workflow.

Keywords　Scientific workflow, Dynamic, Case-based reasoning

1　引言

20世纪90年代初，随着问题求解环境(PSE)在科学研究活动中的应用，科学工作流和科学工作流管理被引入到科学问题求解环境中。在某种意义上，科学工作流管理属于传统的工作流管理的范畴。从共性来说，科学工作流和商业工作流都是对某一过程的描述，只不过科学工作流描述的是科学活动过程，而商业工作流描述的是商业活动过程。但是，二者由于所面向应用领域的差异，各自有不同的特点。在商业活动过程中，工作流管理主要是处理管理数据、财务数据以及一些结构化的文档，比如合同、报表等。因此商业工作流具有静态性和高度的重复性。而在科学活动中，由于科学活动过程往往具有不可预知性，因此，科学工作流是动态的，而不是静态的[1]。科学工作流的定义不可能像商业工作流的定义那样做到对整个过程的完全了解，在某些情况下不可能定义一个完整的科学工作流。虽然科学工作流也有一定的重复性，但是和商业活动中公式化的重复性不同，科学工作流中的重复性往往只是体现在抽象工作步骤和某些具体活动中的重复，重复性远远低于商业工作流的重复性[2,3]。

科学工作流管理是通过科学工作流建模来实现的。和商业领域的科学工作流模型类似，科学工作流模型通过定义任务(活动)、任务间的逻辑关系、数据和资源等对科学工作流流程进行抽象处理，是对科学工作流所描述的科学问题求解过程的一种抽象。虽然这个科学问题的求解过程也遵循一定的基本流程，但是整个过程中的诸多细节却是动态的、不确定的，科学工作流的定义往往是不完整的，在执行过程中也需要根据工作流的运行态势做出动态修改。而传统的工作流模型仅对可预见、可事先给出完整定义的工作流流程进行管理，对流程的动态变化因素缺乏支持[4]。显然，这种传统工作流模型不能满足科学工作流对动态性的要求。

为解决科学工作流的动态性问题，相关研究人员提出了多种解决方法。这些解决方法概括起来主要有以下几种方式：改进动态工作流的建模方式[5,6]、应用动态工作流组件(Dynamic Flow Composition)[7,8]、改进工作流的执行策略(Execution Strategies)[9,10]、提高工作流执行过程中的人-机交互[11,12]、基于Agent智能的自适应控制[13]等。虽然这些方法都能在一定程度上提高科学工作流的动态性，但是从本质上来说，建立一种动态的、自适应性的动态科学工作流模型，使得科学工作流在执行过程中能够根据环境的变化或者用户需求的变化做出有效的变化才能很好地满足科学工作流对动态性的要求。

本文在上述研究的基础上，建立了一种树状结构的动态科学工作流模型。利用这种动态科学工作流模型，可以根据科学活动的特点，在科学工作流的定义中将科学工作流的定义分为高层抽象工作流和低层执行工作流。这种动态科学工作流模型通过与基于案例的推理技术相结合，能很好地解决科学工作流对动态性的要求，提高科学工作流管理系统的自适应性。同时，基于案例推理的重用为解决科学工作流低重复性问题、实现科学工作流从单个计算步骤到整个流程定义的多层次重用提供了有效的解决手段。

2　动态科学工作流模型

为描述模型，首先给出以下定义：

定义 2-1(运行环境)　科学工作流的运行环境是一系列

文元桥　讲师，博士生；**余胜生**　教授，博士生导师。

环境因子函数的集合，表示为

$swf\_env \equiv \{f_1, f_2, \cdots, f_N\}$

其中，$f_i (i \in N)$为环境因子函数，表示的是科学工作流实例在某一时刻的某一个特性。该特性和时间有关，在不同的时间点上，这个特性是不同的；在不同的科学活动中，这个特性也可能不同。

定义 2-2(活动) 科学工作流运行的最小工作单位称为活动，它是资源与操作的结合过程，具体可以表示为

$swf\_act = \{(res\_1, ope\_1), (res\_2, ope\_2), \cdots, (res\_N, ope\_N)\}$

其中，$res_i (i \in N)$表示完成活动所需的某个资源(资源包括软件资源(应用程序、数值模式等)、硬件资源以及人等)i；ope_i 表示对应于资源 i 的一个具体操作。

定义 2-3(活动约束条件) 活动约束条件是科学工作流执行过程中不同活动之间约束关系的集合，表示为

$swf\_con\_al = \{ac_1, ac_2, ac_3, \cdots\}$

其中 ac_i 是活动之间的约束关系，比如因果约束、结果值约束等。

定义 2-4(步骤) 在科学工作流中，科学活动的某一特定阶段称为一个步骤，具体可表示为一个二元组：

$swf\_step \equiv (SWFACT, SWFCON\_A)$

其中，SWFACT 表示活动 swf_act 的集合；SWFCON_A 是工作流执行过程中活动的转移条件 swf_con_a 的集合，swf_con_a 又是工作流运行环境 swf_env 和工作流活动的约束条件 swf_con_al 的函数，即

$swf\_con\_a \equiv (swf\_env, swf\_con\_al)$

定义 2-5(步骤约束条件) 步骤约束条件是科学工作流流程中不同步骤之间约束关系的集合，表示为

$swf\_con\_sl = \{sc_1, sc_2, sc_3, \cdots\}$

其中 sc_i 是步骤之间的约束关系，比如因果约束、结果值约束等。

定义 2-6(流程) 科学工作流的一个具体执行过程称为一个科学工作流流程，它表示为一个二元组：

$swf\_process \equiv (SWFSTEP, SWFCON\_S)$

其中，SWFSTEP 是工作流步骤 swf_step 的集合；SWFCON_S 是工作流步骤转移条件 swf_con_s 的集合，swf_con_s 又是工作流运行环境 swf_env 和流程步骤的约束条件 swf_con_sl 的函数，即

$swf\_con\_s \equiv (swf\_env, swf\_con\_sl)$

根据上述定义，本文建立了一个树状层次结构的动态工作流模型(图 1)。如图 1 所示，在动态科学工作流模型中，步骤隶属于流程，步骤中可以包含其它流程，活动隶属于步骤。工作流的设计采用自顶向下的方法，首先将一个流程分解为多个步骤，然后以步骤为单位定义具体的流程以及活动。步骤的父节点为流程，活动的父节点为步骤。步骤和步骤之间的约束关系使得步骤和步骤节点相互间可能构成父子关系或兄弟关系。同样，活动之间的约束关系使得同一父节点下的活动节点构成父子关系或者兄弟，形成一个树状的完整工作流模型。在这个树状模型中，只有流程可以作为根节点，也可以作为中间节点；步骤只能作为中间节点，不可以作为叶节点；而活动只可以作为步骤的叶节点。这种树状的动态结构模型能够方便用户对工作流进行动态的修改而不致于引起系统的混乱。用户在工作流执行过程中，可以根据需要随时调整步骤和步骤之间的关系或者活动和活动之间的关系，也可以根据需要调整、修改工作流流程的定义，实现对工作流的动态管理和监控。

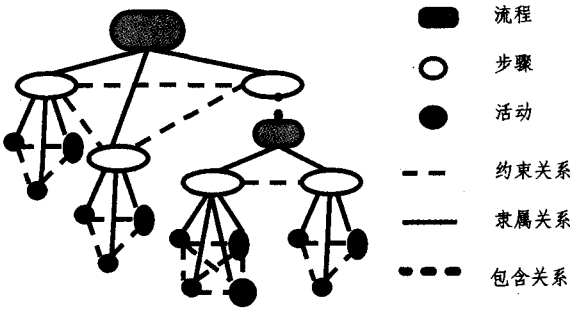

图 1 动态科学工作流模型

3 基于有向图的模型描述

科学工作流流程的定义是一个科学活动过程支持自动或者半自动操作的形式化表现，流程的定义过程实际上是对科学工作流模型的描述过程。在这个过程中，它需要描述组成流程的各个步骤之间的相互关系，也要描述每个步骤中所属的各种活动以及这些活动之间的相互关系。科学工作流模型除了需要能够支持完整的科学工作流的概念定义，为用户提供定义工作所需的组件或者元素等主要特征外，还应该能够适应科学工作流动态变化的特征，满足用户在建模过程中提出的各种要求。本节利用有向图来详细描述上节建立的动态科学工作流模型。

在本文给出的动态科学工作流模型中，步骤节点永远作为父节点出现(也就是说步骤节点不可以作为叶节点)，而活动节点永远作为叶节点出现，因此可以将工作流的定义分为两个层次：第一个层次为高层抽象，即将科学工作流流程划分为相互关联的一些步骤，称之为高层抽象工作流；第二个层次是每个步骤的具体化，也就是描述步骤中所包含的所有活动之间的关系，称之为低层执行工作流。之所以将科学工作流的定义划分为多个层次，一方面是考虑科学活动过程中研究人员的思维习惯，另一方面是考虑到科学工作流的动态性。在科学活动过程中，研究人员往往都要经历一个从抽象到具体的过程，而且需要根据活动的进展对后续的活动做一些必要的调整。

在有向图中，建立了三种节点和三种有向线，即任务节点、控制节点、资源节点；数据有向线、控制有向线、资源有向线，分别用 TN、CN、RN 和 DDL、CDL、RDL 表示三种节点和三种有向线的集合，其中

$TN = \{tn_1, tn_2, tn_3, \cdots\}$，表示任务节点集合，集合中的每个元素表示一个的独立的具体任务，这个任务在不同情况下是有区别的。在高层工作流中，一个任务节点就表示一个具体的步骤所包含的任务。由于在动态模型中允许步骤包含其它流程，因此高层抽象工作流中的任务节点又分为组合任务节点和原子任务节点。所谓组合任务节点，是指包含其它流程的任务节点，该节点可以被划分为多个原子任务节点。而原子任务节点是指不可再分的任务节点。在底层工作流中，一个任务节点代表一个具体活动所包含的任务，它代表了最小任务单位，不可再分，又称为活动任务节点。

$CN = \{cn_1, cn_2, cn_3, \cdots\}$，代表所有任务之间的逻辑关系的集合，集合中的每个元素就是一个控制节点。控制节点通过执行一种控制操作(与操作(AND)、或操作(OR)、顺序操

作(SEQ)中的一种)决定任务节点之间的执行逻辑关系,也就是路由结构。路由结构是流程执行路线的依据,根据科学计算过程的特点和科学工作流管理的需求,本文在在有向图中定义了如表1所示的5种基本路由结构。

$RN=\{rn_1, rn_2, rn_3, \cdots\}$,代表所有与任务相关的资源集合,每个元素代表一个具有一定功能的资源,包括应用程序、数值模式、数据库/数据集等。

$CDL \subseteq TN \times CN \cup CN \times TN$,代表由任务间的约束关系定义的控制流有向线的集合,表示任务点和控制节点之间的逻辑关系。

$DDL \subseteq TN \times TN$,代表由任务的输入输出定义的数据流有向线的集合,表示任务节点之间数据的流向。

$RDL \subseteq RN \times TN$,代表由完成任务所需资源定义的资源流有向线的集合,表示活动对资源的需求。

对于任务节点,还必须满足以下约束关系:

(1) 对于任何一个任务节点,必须至少与一个控制节点相连。即

$\forall tn_i \in TN, \exists cn_j \in CN$

其中$(tn_i, cn_j) \in CDL \vee (cn_j, tn_i) \in CDL$

(2) 如果某个任务节点 tn_s 满足条件

$\exists!\ tn_s \in TN, \exists cn_m \in CN, (cn_m, tn_s) \notin CDL$

则称 tn_s 为起始节点。

(3) 如果某个任务节点 tn_e 满足条件

$\exists!\ tn_e \in TN, \forall cn_m \in CN, (tn_e, cn_m) \notin CDL$

则称 tn_e 为终止节点。

表1 基本路由结构表

基本路由结构类型	控制操作类型	定 义
并行分支(AND-Split)	与(AND)	如果一个任务的后继的两个或多个任务需要并行执行,就采用并行分支,见图2(A)。
并行连接(AND-Join)	与(AND)	如果有多个并行分支汇聚到一个控制节点上并且要等到各个分支都执行完才能执行后续任务,就采用并行连接,见图2(B)。
选择分支(OR-Split)	或(OR)	如果一个任务要根据某种选择规则从两个或者两个以上的后续任务中选择一个或者多个任务来执行,就采用选择分支,见图2(C)。
选择连接(OR-Join)	或(OR)	如果一个控制节点可能有一个或多个分支被激活,那么要等待所有被激活的分支都执行完,才能执行后续任务,就采用选择连接,见图2(D)。
顺序(Sequence)	顺序(SEQ)	如果两个或多个任务按照固定顺序串行地执行,就采用顺序结构,见图2(E)。

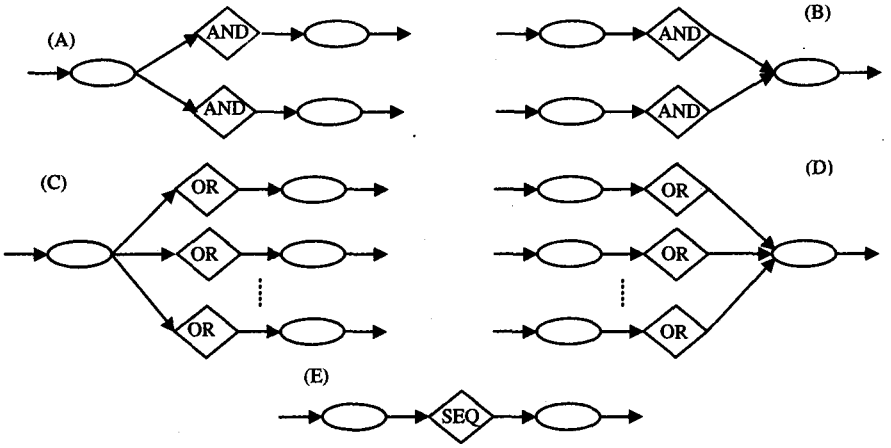

图2 基本路由结构示意图

对于高层抽象工作流,主要目的是将工作流过程抽象为步骤,因此其任务就是建立步骤之间的逻辑关系。由于不涉及到具体的资源,只需要定义两种节点(任务节点和控制节点)和两种有向线(数据有向线和控制有向线)。在有向图描述中,一个工作流流程被表达为一个四元组:

swf_process_h≡(TN,CN,CDL,DDL)

对于低层执行工作流,主要是对一个步骤内所有活动的具体描述,因此需要定义任务节点、控制节点和资源节点3种节点和数据有向线、控制有向线和资源有向线3种有向线。在低层工作流定义有向图描述中,一个工作流被表达为一个六元组:

swf_process_l≡(TN,CN,RN,CDL,DDL,RDL)

图3展示了动态科学工作流模型的有向图表示。如图3所示,一个科学活动首先被抽象为3个步骤,分别用3个任务节点 tn1、tn2、tn3 表示。其中,tn1 包含两个子任务(用 tn(1,1)和 tn(1,2)表示),tn2 包含两个子任务(用 tn(2,1)和 tn(2,2)表示),tn3 包含三个子任务(用 tn(3,1)、tn(3,2)和 tn(3,3)表示)。在所有的子任务节点中,tn(3,2)和 tn(3,3)属于组合任务节点,其它属于原子任务节点。原子任务节点直接与低层活动任务节点对应,而组合任务节点则对应于多个相互作用的活动任务节点,tn(3,2)对应于 tn(3,2,1)、tn(3,2,2)、tn(3,2,3)、tn(3,2,4),tn(3,3)对应于 tn(3,3,1)、tn(3,3,2)、tn(3,3,3)。工作流管理系统最终执行的是由所有活动任务节点组成的工作流。

图3所示的低层执行工作流可以用六元组描述如下:

swf_process_l≡(TN,CN,RN,CDL,DDL,RDL)

其中,TN={tn(1,1),tn(1,2),tn(2,1),tn(2,2),tn(3,1),tn(3,2,1),tn(3,2,2),tn(3,2,3),tn(3,2,4),tn(3,3,1),tn(3,3,2),tn(3,3,3)}

CN={cn1, cn2, cn3, cn4, cn5, cn6, cn7, cn8}

RN={rn1, rn2, rn3, rn4, rn5, rn6, rn7, rn8, rn9, rn10}

CDL={(tn(1,1),cn1),(cn1,tn(1,2)), (tn(1,2),cn2), (cn2,tn(2,1)), (cn2,tn(2,2)),(tn(2,1),cn3), (tn(2,2),cn3),(cn3,tn(3,1)), (tn(3,1),cn4), (cn4,tn(3,2,1)),(tn(2,2,1),cn5),(cn5,tn(3,2,2)), (cn5,tn(3,2,3)), (tn(3,2,2),cn6),(tn

(3,2,3),cn6),(cn6,tn(3,2,4)),(tn(3,2,4),cn7),(cn7,tn(3,3,1)),(cn7,tn(3,3,2),(tn(3,3,1),cn8),(tn(3,3,2),cn8),(cn8,tn(3,3,3))}

DDL={(tn(1,1),tn(1,2)),(tn(1,2),tn(2,1)),(tn(1,2),tn(2,2)),(tn(2,1),tn(3,1)),(tn(2,2),tn(3,1)),(tn(3,1),tn(3,2,1)),(tn(3,2,1),tn(3,2,2)),(tn(3,2,1),tn(3,2,3)),(tn(3,3,2),tn(3,2,4)),(tn(3,2,3),tn(3,2,4)),(tn(3,2,4),tn(3,3,1)),(tn(3,2,4),tn(3,3,2)),(tn(3,3,1),tn(3,3,3)),(tn(3,3,2),tn(3,3,3))}

RDL={(rn1,tn(1,1)),(rn2,tn(1,2)),(rn3,tn(2,1)),(rn3,tn(2,2)),(rn4,tn(3,1)),(rn5,tn(3,2,1)),(rn6,tn(3,2,2)),(rn6,tn(3,2,3)),(rn7,tn(3,2,4)),(rn8,tn(3,3,1)),(rn9,tn(3,3,2)),(rn10,tn(3,3,3))}

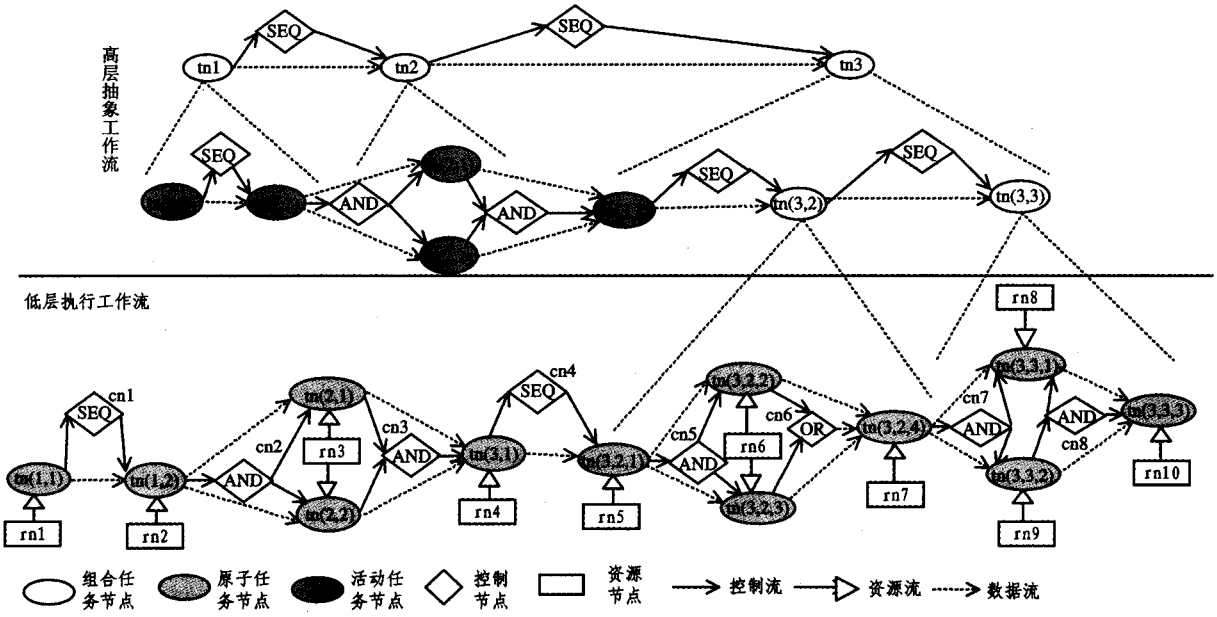

图 3　动态科学工作流模型的有向图表示

考虑到科学工作流流程定义的数据有易存储性和通用性，在基于动态科学工作流模型的科学工作流管理系统中，通过 XML 语言来描述科学工作流流程模型的定义。在 XML 中，利用有向图描述的流程模型包含以下元素：任务节点和流程。流程被抽象为高层抽象工作流和低层执行工作流，其数据结构分别由 swf_process_h 和 swf_process_l 给出。对于任务节点，其数据结构被描述为一个八元组：

TN≡(P_id, S_id, TN_id, C_{pr}, C_{post}, $Input$, $Output$, $Operation$)

元组中各元素的意义如下：

P_id：任务节点所属流程的唯一标识符(identifier)；

S_id：任务节点所属步骤的唯一标识符；

TN_id：任务节点在所属步骤中的唯一标识符；

C_{pr}：任务节点与前续控制节点之间逻辑关系的集合，可以表示为

C_{pr} = {(pr_cn_1, pr_logic_1), (pr_cn_2, pr_logic_2),…}

其中，pr_cn_i 表示与任务节点相连第 i 个前续控制节点；pr_logic_i 表示任务节点与前续第 i 个控制节点之间的逻辑关系，为 AND、OR 或者 SEQ 中的一种，分别用 AND_IN，OR_IN，SEQ_IN 表示。对于起始节点，则标示为 STAT。

C_{post}：任务节点后续控制节点之间逻辑关系的集合，可以表示为

C_{post} = {(po_cn_1, po_logic_1), (po_cn_2, po_logic_2),…}

其中，po_cn_i 表示与任务节点相连的第 i 个后续控制节点；po_logic_i 表示任务节点与后续第 i 个控制节点之间的逻辑关系，为 AND、OR 或者 SEQ 中的一种，分别用 AND_OUT，OR_OUT，SEQ_OUT 表示。对于终止节点，则标示为 END。

$Input$：任务节点输入数据的集合，可以表示为

$Input$ = {($inputNode\_1$, $input\_1$), ($inPutNode\_2$, $input\_2$),…}

其中，$inPutNode\_i$ 表示第 i 个向任务节点输入数据的节点(可以是任务节点和资源节点)；$input\_i$ 表示第 i 个向任务节点输入数据的节点所输入的数据。

$Output$：任务节点输出数据的集合，可以表示为

$Output$ = {($outputNode\_1$, $output\_1$), ($outputNode\_2$, $output\_2$),…}

其中，$OutputNode\_i$ 表示第 i 个接受任务节点输出数据的节点(可以是任务节点和资源节点)；$output\_i$ 表示第 i 个接受任务节点输出数据的节点所接收的数据。

$Operation$：任务节点所需操作的集合，可以表示为

$Operation$ = {($resource\_1$, $action\_1$), ($resource\_2$, $action\_2$),…}

其中，$resource\_i$ 表示某个资源，而 $action\_i$ 表示对应资源($resource\_i$)的一个操作行为。

上述 7 个元素详细地定义了任务节点的控制条件、数据依赖性、与其它任务节点之间的逻辑关系以及所需资源和对应资源的操作，使得任务节点具有良好的独立表达能力。在实际工作流的执行中，每个任务节点都可以作为一个单独的单元实现分布式环境下的执行。对于整个系统来说，这种独立的基于任务节点的定义使得工作流具有良好的动态性和重用性。用户可以根据需要，动态调度和调整每个任务节点的

运行，也可以实现原子级的重用，适应了科学工作流动态性和低重复性的需求。

4 基于案例推理的重用

为满足科学活动动态性的要求，实现最大程度上的重用，本文采用了基于案例的推理方法[14,15]，来实现从单个计算步骤到整个计算流程的不同层次的重用。

定义 4-1(科学工作流案例) 在科学工作流管理系统中，系统对每一个科学工作流运行过程的详细记录形成一个科学工作流案例。

定义 4-2(科学工作流流程相似度) 科学工作流流程相似度表示两个科学工作流案例(记为 I、S)工作流流程定义之间的相似度，简称为流程相似度，记为 $Sim(I,S)\_process$。

定义 4-3(科学工作流步骤相似度) 科学工作流步骤相似度表示两个科学工作流案例(记为 I、S)中不同工作流步骤定义之间的相似度，简称为步骤相似度，记为 $Sim(I,S)\_step$。

流程相似度 $Sim(I,S)\_process$ 和步骤相似度 $Sim(I,S)\_step$ 由下式计算：

$$Sim(I,S)=\sum_{i=1}^{N} w_i \times f_i(\vec{I}_i,\vec{S}_i)$$

其中，w_i 为系统常数，且有 $\sum_{i=1}^{N} w_i=1$；f_i 为相似函数，定义为

$$f_i(\vec{I}_i,\vec{S}_i)=\frac{|m|}{\max(|\vec{I}_i|,|\vec{S}_i|)}\times\sum_{I_i^k\Delta S_i^j}\frac{1}{k}\left(1-\frac{|\vec{I}_i^k-\vec{S}_i^j|}{||\vec{I}_i^k|+|\vec{S}_i^j|+1|}\right)$$

其中 $\vec{I}_i$ 和 $\vec{S}_i$ 分别为流程相似度或步骤相似度的第 i 个特征向量；$|\vec{I}_i|$、$|\vec{S}_i|$ 分别表示向量 $\vec{I}_i$ 和 $\vec{S}_i$ 的维数；假设 $\vec{I}_i$ 和 $\vec{S}_i$ 有 m 个因子定义相同；$I_i^k\Delta S_i^j$ 表示 $\vec{I}_i$ 的第 k 个因子和 $\vec{S}_i$ 的第 j 个因子定义相同；I_i^k 和 S_i^j 分别代表 $\vec{I}_i$ 的第 k 个因子和 $\vec{S}_i$ 的第 j 个因子。

在本文中，根据科学工作流的模型的定义，在流程相似度的计算中，取 SWFSTEP 和 SWFCON_S 为相似度计算的特征向量，则流程相似度的计算为

$$Sim(I,S)\_process=w_1 f(step\_I,step\_S)+w_2 f(scon\_I,scon\_S)$$

其中，$w_1+w_2=1$，$f(step\_I,step\_S)$ 为 I、S 两个案例流程中步骤之间的相似度函数；$f(scon\_I,scon\_S)$ 为两个案例流程中步骤之间约束条件之间的相似度函数。

在步骤相似度的计算中，取 SWFCON_A 和 SWFACT 为相似度计算的特征向量，则步骤相似度的计算可表示为

$$Sim(I,S)\_step=w_1 f(acon\_I,acon\_S)+w_2 f(ares\_I,ares\_S)+w_3 f(aope\_I,aope\_S)$$

其中 $w_1+w_2+w_3=1$，为 $f(acon\_I,acon\_S)$ I、S 两个案例相似步骤中各个活动之间约束条件之间的相似度函数；$f(ares\_I,ares\_S)$ 为两个案例相似步骤中各个活动所需资源之间的相似度函数；$f(aope\_I,aope\_S)$ 为两个案例相似步骤中与各个活动所需资源相对应的操作之间的相似度函数。

在科学工作流管理系统中，在每个工作流流程的执行过程中，系统不仅仅记录了每个工作流流程的定义，而且对工作流流程中每个步骤的运行也会详细记录。当系统运行一段时间后，就形成了多个科学工作流案例。当用户定义一个新的科学工作流时，可以利用基于案例的推理方法实现不同层次的重用。具体算法设计如下：

算法 1 工作流流程定义重用(假设拟输入的科学工作流流程为 $process\_I$)

步骤 1：假设系统存储有 K 个科学工作流案例，分别为 $S^1,S^2,\cdots,S^K$；

步骤 2：计算 $Sim(I,S^k)\_process$，其中 $k=1,2,\cdots,K$；

步骤 3：假设 $Sim(I,S)\_process^{max}$ 是 $Sim(I,S^k)\_process$ 中的最大者，而且 $Sim(I,S)\_process^{max}\geqslant Sim(I,S)\_process^{val}$，其中 $Sim(I,S)\_process^{val}$ 为工作流案例流程相似度的阈值(由用户设定)，输出 $Sim(I,S)\_process^{max}$ 所对应的科学工作流案例的工作流流程定义，并返回。

算法 2 工作流步骤重用(假设拟输入的科学工作流步骤为 $step\_I$，其所属的科学工作流流程为 $process\_I$)

步骤 1：假设系统存储有 K 个科学工作流案例，分别为 $S^1,S^2,\cdots,S^K$；

步骤 2：计算 $Sim(I,S^k)\_process$，其中 $k=1,2,\cdots,K$；

步骤 3：假设 $Sim(I,S^k)\_process$ 中有 M 个不小于 $Sim(I,S)\_process^{val}$ 的案例，记为 $Sim(I,S)\_process^m$，其中 $Sim(I,S)\_process^m\geqslant Sim(I,S)\_process^{val}$，$m=1,2,\cdots,M$，$M\leqslant K$。计算 $Sim(I,S)\_process^m$ 所对应的 M 个案例的 L 个步骤与 $step\_I$ 之间的相似度 $Sim(I,S)\_step_m^l(l=1,2,\cdots,L)$。

步骤 4：假设 $Sim(I,S)\_step^{max}$ 是 $Sim(I,S)\_step_m^l$ 中的最大者，而且 $Sim(I,S)\_step^{max}\geqslant Sim(I,S)\_step^{val}$，其中 $Sim(I,S)\_step^{val}$ 为工作流案例步骤相似度的阈值(由用户设定)，输出 $Sim(I,S)\_step^{max}$ 所对应的科学工作流案例的工作流步骤的详细定义，并返回。

小结 本文根据科学活动的特点，为满足科学工作流动态性的要求，提出了一种基于树状结构的动态科学工作流模型，这种树状的动态结构模型将科学工作流的设计分为了高层抽象工作流和低层执行工作流，能够方便用户采用自顶向下的方法设计不完整的工作流。在工作流执行过程中，用户可以根据需要对科学工作流进行动态调整，实现对工作流的动态监控和管理。基于案例推理的重用机制，可以实现从一个科学工作流步骤到一个完整的科学工作流流程定义的重用，从而使系统具有了一定的自适应性，为科学工作流低重复性问题提供了一种良好的解决手段。

本文虽然对动态工作流模型进行了详细的讨论，但是对于模型中一些规则、算法的合理性、完整性没有给予严格的证明，这将是进一步研究的内容。

参 考 文 献

1 Zhao Zhiming, Belloum A, Sloot P, et al. Agent Technology and Scientific Workflow Management in an E-Science Environment. In: Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), 2005. 19～23

2 Medeiros C, Vossen G, Weske M. WASA: A Workflow-based Architecture to Support Scientific Database Applications. In: Proceedings of the 6th DEXA Conference. London, England, 1995. 574～583

3 Ailamaki A, Ioannidis Y E, Livny M. Scientific Workflow Management by Database Management. In: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM 1998). 1998. 190～199

4 van der WMP A, Jablonski S. Dealing with Workflow Change: Identification of Issues and Solutions. International Journal of Computer Systems, and Engineering, 2000, 15(5): 267～276

(下转第 143 页)

关应用不断加入传统的网络中，网络的动态性和不可确定性大大提高。在动态网络中，最常见的就是网络拓扑的动态变化。在网络的运行中，拓扑结构经常变化，不断有新的结点加入或者现有的结点退出，还有结点之间的连接关系也会发生改变。结点之间的连接的改变既包括物理的改变也包括逻辑的改变。目前所研究的 Ad-hoc 网络与 Sensor 网络其实就是一种典型的动态拓扑网络。另外，当前的传统网络的拓扑结构也是经常变化的。因此，建立适应网络拓扑变化的合作与协调模型[25,26]是多 Agent 系统应用于复杂环境首先面临和需要解决的基本问题之一。

结束语 多 Agent 系统(MAS)是综合社会学、经济学和计算机科学等多门学科的交叉研究领域。合作与协调一直是多 Agent 系统研究的关键技术之一。然而，目前提出的一些合作和协调机制可能比较偏向于某一类应用，缺乏通用性。特别是在未来动态、开放、异构的环境下，任何一类合作与协调机制可能都不能解决所有的冲突和不一致。合作与协调需要以上 3 类机制的充分结合。通过事前规划，充分考虑子任务之间的约束依赖关系，尽可能减少冲突和不一致出现的可能性。复杂的环境可能在规划执行中出现突发状况，这时需要多个 Agent 的协商交互，从而达到协调一致的行为。对于某些不一致，可以采用缓慢退化的合作与协调，容忍一定程度的不一致，而减少通信和同步等方面的耗费。此外，对于大规模的多 Agent 系统，上述机制在效率方面都存在明显不足之处。因此，结合人工智能和分布式系统最新的研究成果，综合 3 类机制的优点，开发高效的、通用的合作与协调机制是未来的研究目标和方向。

参 考 文 献

1 Pieter B, Adriaan M, Jeroen Y, et al. Coordinating self-interested planning agent. Autonomous Agents and Multi-Agent System, 2006,12:199～218
2 Javier V S, Virginia D, Frank D. Organizing multi-agent systems. Autonomous Agents and Multi-Agent System, 2005,3:307～360
3 Wooldridge M. An Introduction to MultiAgent System. John Wiley & Sons, 2002
4 Sycra K. Multiagent systems. AI Magazine, 1998,19(2): 79～92
5 Jennings N R, Sycrra K, Wooldridge M. A roadmap of agent research and development. Automous Agents and Multi-Agent System, 1998, 1(1):7～38
6 Noriega P, Sierra C. Agent Mediated Electronic Commerce. Lecture Notes in Artificial Intelligence 1571, Springer, 1999
7 Kuokka D R, Harada L P. Issues and extensions for information matchmaking protocols. International Journal of Cooperative Information System, 1996, 5 (2-3): 251～274
8 Davis R, Smith R G. Negotiation as a metaphor for distributed problem solving. Artificial Intelligence, 1983, 20:63～100
9 Wooldridge M, Jennings N R. Intelligent agents: theory and practice. The Knowledge Engineering Review, 1995, 10 (2): 115～152
10 Russell S J, Norvig P. Artificial Intelligence: a Modern Approach. 2nd edition. Prentice Hall, 2003
11 Smith R G, Davis R. Frameworks for cooperation in distributed problem solving. IEEE Transactions on Systems, Man and Cybernetics, 1980,11 (1)
12 Georgeff M. Communication and interaction in multi-agent planning. In: National Conference Artificial Intelligence, 1983. 125～129
13 Georgeff M. A theory of action for multi-agent planning. In: National Conference Artificial Intelligence, 1984. 121～125
14 Corkill D D. Hierarchical planning in a distributed environment. In: National Conference Artificial Intelligence, 1979. 168～179
15 Rosenschein J S, Genesereth M R. Communication and cooperation among logic-based agents. In: Proceedings of Computer Communication, 1987. 594～600
16 Lesser V R. A Retrospective View of FA/C Distributed Problem Solving. IEEE Transaction on Systems, Man and Cybernetics, 1991,21 (6): 1347～1362
17 Lesser V R, Corkill D D. Functionally accurate, cooperative distributed systems. IEEE Transaction on Systems, Man and Cybernetics, 1981,SMC-11:81～96
18 Durfee E H. Planning in distributed artificial intelligence. In: Foundations of Distributed Artificial Intelligence, 1996. 231～245
19 Parsons S, Sierra C A, Jennings N R. Agents that reason and negotiate by arguing. Journal of Logic and Computation, 1998, 8 (3): 261～292
20 Fox J, Krause P, Ambler S. Arguments, contradictions and practical reasoning. In: Proceedings of the 10th European Conf on Artificial Intelligence, 1992. 623～627
21 Rosenschein J S. Rational Interaction: Cooperation among Intelligent Agents: [PhD thesis]. CA 4305. Computer Science Department, Stanford University, 1985
22 Gmytrasiewicz P J, Durfee E H. Rational Coordination in Multi-agent Environments. Autonomous Agents and Multi-agent Systems, 2000, 3:319～350
23 Shoham Y, Tennenholtz M. Emergent conventions in multi-agent systems. In: Proceedings of Knowledge Representation and Reasoning, 1992. 225～231
24 Shoham Y, Tennenholtz M. On social laws for artificial agent societies: off-line design. In: Proceedings of Knowledge Representation and Reasoning, 1992. 225～231
25 Jiang Y C, Jiang J C. A Multi-agent Coordination Model for the Variation of Underlying Network Topology. Expert Systems with Applications (Elsevier Science), 2005, 29(2): 372～382
26 Jiang Y C, Xia Z Y, Zhang S Y. An Adaptive Adjusting Mechanism for Agents Distributed Blackboard Architecture. Microprocessors and Microsystems, 2004, 29(1):9～20
27 王玥，陈世福. 基于多 Agent 的 Teamwork 研究综述. 计算机科学，2002, 29(10): 38～42
28 李静，陈兆乾，陈世福，等. 多 Agent Teamwork 研究综述. 计算机研究与发展，2003, 40(3):422～429
29 李海刚，吴启迪. 多 Agent 系统研究综述. 同济大学学报，2003, 31(6):728～732
30 李庆华，张红君. 开放 Agent 社会的框架模型研究综述. 计算机科学，2005,32(7):137～141

(上接第 124 页)

5 Bogia D P, Kaplan S M. Flexibility and Control for Dynamic Workflows in the Worlds Environment. In: Proceedings of conference on Organizational Computing Systems. New York, USA, ACM Press, 1995. 148～159
6 Jorgensen H D. Interaction as a Framework for Flexible Workflow Modelling. In: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work. Singapore, ACM Press, 2001. 32～41
7 Bubak M, Gubala T, Kapalka M, et al. Grid Service Registry for Workflow Composition Framework. In: Proceedings of International Conference on Computational Science. LNCS 3038, Springer, 2004. 34～41
8 Caragea D, Syeda-Mahmood T. Semantic API Matching for Automatic Service Composition. In: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters. New York, USA, ACM Press, 2004. 436～437
9 Baggio G, Wainer J, Ellis C. Applying Scheduling Techniques to Minimize the Number of Late Jobs in Workflow Systems. In: Proceedings of the 2004 ACM Symposium on Applied Computing. New York, USA, ACM Press, 2001. 1396～1403
10 Jin L jie, Casati F, Sayal M, et al. Load Balancing in Distributed Workflow Management System. In: Proceedings of the 2001 ACM Symposium on Applied Computing. New York, USA. ACM Press. 2001. 522～530
11 Oinn T, Addis M, Ferris J, et al. Taverna: A tool for the Composition and Enactment of Bioinformatics Workflows. Bioinformatics Journal, 2004, 20(17): 3045～3054
12 Afsarmanesh H, Belleman R, Belloum A, et al. VLAM-G: A Grid-based Virtual Laboratory. Scientific Programming: Special Issue on Grid Computing, 2002, 10(2): 173～181
13 Zhao Z, Belloum A, Yakali H, et al. Dynamic Workflow in a Grid Enabled Problem Solving Environment. In: Proceedings of the 5th International Conference on Computer and Information Technology (CIT2005). Shanghai, China. IEEE Computer Society Press, 2005. 339～345
14 Kaster D S, Mediros C B, Rocha H V. Supporting Modeling and Problem Solving from Precedent Experience: the Role of Workflows and Case-Based Reasoning. Environmental Modeling & Software, 2005, 20:689～704
15 顾大可，俞勇. 一个基于案例的动态工作流模型. 上海交通大学学报，2001,35(9):1290～1292