

DOI:10.3969/j.issn.1673-4785.201210067
网络出版地址: <http://www.cnki.net/kcms/detail/23.1538.TP.20130929.1042.002.html>

水下机器人任务流程建模与管理

张鸿皓,姜大鹏,庞永杰,李金龙

(哈尔滨工程大学 智能水下机器人技术国防科技重点实验室,黑龙江 哈尔滨 150001)

摘 要:为实现水下机器人任务流程的快速建模与自动管理,通过改进传统的工作流理论,提出了一种实用的任务流程模型——任务工作流.介绍了作为任务工作流理论基础的传统 Petri 网及工作流模型,指出了原有理论的缺陷并介绍了任务工作流的基本理论.以任务工作流模型为基础,实现了一款任务流程管理系统,为其定义了完善的图形化建模手段与标准的文本化的描述语言,并建立了一套担负着运行时任务管理工作的核心应用程序,能够实现任务的描述与建模、自动运行与管理,以及任务流程的动态更改.实践表明,任务工作流建模简单快捷、方便编程实现,是机器人任务建模与管理的理想工具.

关键词:工作流;任务工作流;任务流程管理;Petri 网;水下机器人

中图分类号:TP301.1 **文献标志码:**A **文章编号:**1673-4785(2013)05-0433-06

中文引用格式:张鸿皓,姜大鹏,庞永杰,等.水下机器人任务流程建模与管理[J]. 智能系统学报, 2013, 8(5): 433-438.
英文引用格式:ZHANG Honghao, JIANG Dapeng, PANG Yongjie, et al. Workflow modeling and management on AUV[J]. CAAI Transactions on Intelligent Systems, 2013, 8(5): 433-438.

Workflow modeling and management on AUV

ZHANG Honghao, JIANG Dapeng, PANG Yongjie, LI Jinlong

(State Key Laboratory of Autonomous Underwater Vehicle, Harbin Engineering University, Harbin 150001, China)

Abstract: In order to implement fast modeling and automatic management of the task flow for the autonomous underwater vehicle (AUV), a practical task flow model named “task workflow”, has been proposed to improve the traditional workflow theory. In this paper, the traditional Petri net and workflow model as the theoretical basis of the task workflow are briefly introduced, the defects of the original theory are pointed out and the basic theory of the task workflow is introduced; with the task workflow model as the basis, a task flow management system is realized, the perfect graph modeling means and standard text description language are defined, and a set of core application programs undertaking the task management in the operation are established. The programs may realize task description and modeling, automatic operation and management and the dynamic amendment of task flow. Past experience has shown that task workflow is easy for modeling and programming, and it is an ideal tool for task modeling and management.

Keywords: workflow; task workflow; task flow management; Petri net; AUV

自主机器人的作业流程总是可以在某一层面上划分成多个子过程.宏观的过程可以由子过程、子过程的发生次序以及决定这种次序的条件构成.人们创造了各式各样的规则,用来描述和管理自然发生或人为控制的过程.应用在机器人上的作业流程主要体现人的智能(操控者或机器本身的智能).在机器人相关领域内,对机器细微动作的精确控制早已

实现,但针对体现高度智能的宏观任务流程的描述与自动管理,尚缺乏成熟的理论和实例.

自治式水下机器人由于工作环境与人隔绝,特别需要一套能完整表述任务规划者意愿,并能交由计算机自动执行管理的理论模型及实用化的方法.传统的 Petri 网和工作流模型虽能部分解决上述问题,但多少存在建模困难、编程实现复杂及描述能力受限等问题^[1],尚未出现一套成功而方便实用的解决方案.

任务工作流模型较大幅度地改进了传统的工作流理论,在增强了模型描述能力的同时,也大幅简化

收稿日期:2012-10-31. 网络出版日期:2013-09-29.
基金项目:国家自然科学基金资助项目(51209051);中国博士后科学基金资助项目(2012M520708);中央高校基本科研业务费专项基金资助项目(HEUCFR1203).
通信作者:张鸿皓. E-mail: zhh094@126.com.

了模型的复杂度,能以图形化方式建模并方便地转换为计算机可以理解的文本描述方式,为理论模型的编程实现提供了良好的条件.在此理论基础上构建的水下机器人任务建模与管理系统的已经过实践检验,能为复杂逻辑流程的快速建模与自动管理提供极大的便利.

1 Petri 网

Petri 网由表示状态的元素“库所”(place)和表示变化的元素“变迁”(transition)组成.联系 S 和 T 的是两者间的流关系,用 F 表示.库所可以容纳标记(token),标记代表网路中流动的资源,资源的流动由流关系规定.Petri 网结构是固定的,而库所中标记的分布是可变的,标记因变迁而形成的不同分布表征了 Petri 网的不同状态.

水下机器人的任务建模并不困难,只需用到 Petri 网的基本概念^[2-3],如图 1 及表 1 中的例子所示.

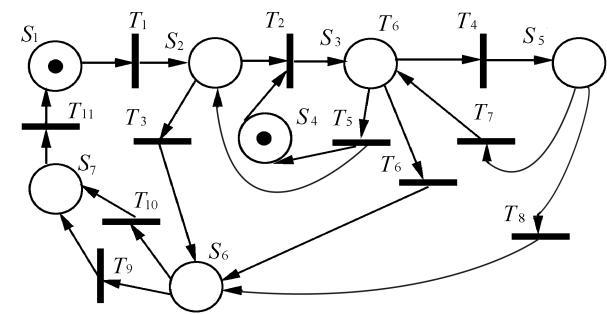


图 1 水下机器人任务示例 (Petri 网)

Fig.1 An example of AUV task flow (Petri net)

表 1 库所与变迁的含义

Table 1 Meanings of places and transition	
库所含义	变迁含义
S_1 : 任务开始	T_1 : AUV 布放
S_2 : 布放结束	T_2 : (成功)驶向目标区域
S_3 : 向目标航行结束	T_3 : (失败)返回回收位置
S_4 : 重规划成功	T_4 : (成功)执行作业任务
S_5 : 作业任务结束	T_5 : 目标点改变,重新规划
S_6 : 到达回收位置	T_6 : 无法到达目标区,返航
S_7 : 回收完成	T_7 : 作业任务重规划
—	T_8 : 作业结束,返航
—	T_9 : 正常回收
—	T_{10} : 回收出现问题或长时间未反应,紧急上浮
—	T_{11} : 加载新的任务指令

图 1 中的一些库所会产生冲突,例如当 S_2 得到一个标记时, T_2 、 T_3 不能同时发生.解决冲突的办法是由环境提供信息,决定由谁来占用资源.所以,决

定了任务路由的一些重要条件实际是无法在 Petri 网中直接表示的,它需要对 Petri 网进行扩展,使标记和网路能够携带和处理更多的环境信息.环境信息的获取和处理往往相当抽象和复杂,通常只能由高级程序语言编程实现.如果要用 Petri 网来描述,由于 Petri 网过于严谨和形式化,将不得不增加一系列的扩展,增加了网系统的复杂程度,使得建模和理解都变得更加困难.

Petri 网只描述事物间的依赖关系,没有更高层的控制流程和其他的人为因素影响,更适合用来描述符合自然规律的自然过程.然而水下机器人作为人类智能乃至机器智能的执行者,其作业过程无疑更接近人为过程.故虽然有人将 Petri 网引入水下机器人的任务管理,但描述和理解困难,不易编程实现,不适合作为抽象任务的建模工具.

2 workflow 模型

workflow 模型是对工作业务过程的抽象表示. workflow 的应用通常需要完成业务过程的计算机化,其模型不仅要让人(不仅是专家,还包括普通用户)容易读懂,也必须让计算机能够理解.

在库所与变迁的基础上,为了定义出顺序、并行、选择、循环等常用的过程逻辑, workflow 网构造了相应的结构化组件,为其规定了特殊的符号,供用户在建模时直接使用^[4].每一组件均可给出对应的 Petri 网建模方法:1) 顺序;2) 并行,它需要 2 个基本的工作流执行原语:“与分支”(AND_split)和“与连接”(AND_join),也可将其当做特殊的任务. AND_split 能同时发起多个任务,而 AND_join 能对多个并行的流程进行同步;3) 选择,根据具体的执行情况进行“多选一”或“多选多”,构造选择路由需要“或分支”(OR_split)和“或连接”(OR_join);4) 循环.

对于 AND_split、AND_join、OR_split、OR_join 这样经常使用的结构,采用图 2 中特定的符号来表示.

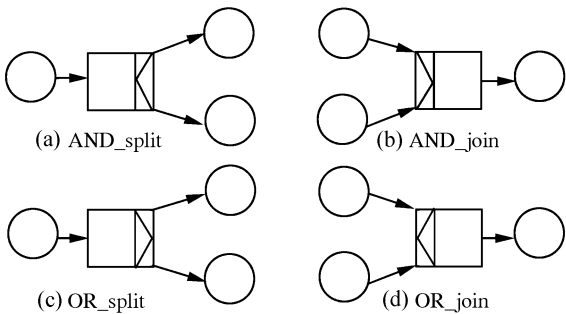


图 2 workflow 逻辑构件

Fig.2 Logical components of workflow

图 3 中的例子与图 1 中的流程大体相似^[5-6].尽管保留了库所、标记等元素,但它并不使用复杂的路径、资源的冲突等技巧来表达逻辑上的意义,而是用固有的逻辑组件来构造路由,将 Petri 网构造的逻辑结构隐藏在逻辑组件之下.

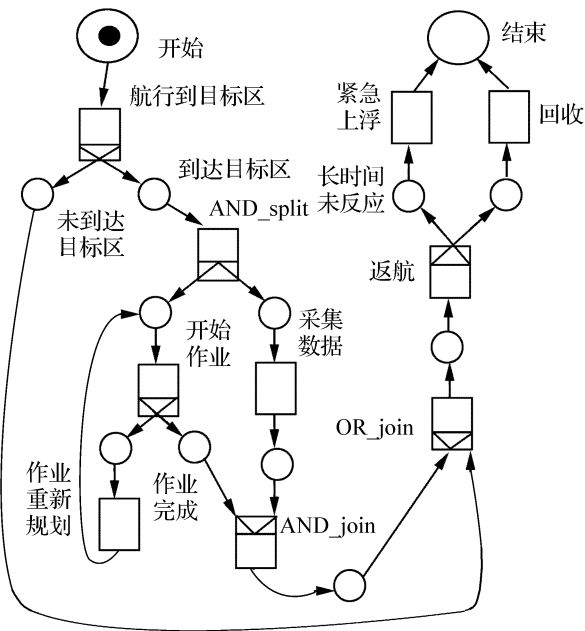


图 3 任务示例(工作流模型)

Fig.3 Example of AUV task flow (workflow)

虽然 Petri 网能够构造出一些精妙而准确的逻辑结构,但在多数时候,这种极其苛求洞察力和智力的建模方式并不适合面向普通人的、需要迅速作出灵活更改的任务建模.与经典的 Petri 网相比,工作流模型更灵活,更接近人类语言和人抽象的思维方式.但同时,由于缺乏标准、精确和成熟的理论基础,虽然市面上的工作流模型及基于工作流的应用软件种类繁多,但每种模型都或多或少存在明显的缺陷,例如 OR_join 任务同步时出现的逻辑问题^[1]等,尚未见到理论上完美的解决方案^[7].然而总的说来,在适当的改进和处理之后,工作流模型能够更好地服务于机器人抽象任务建模.

3 任务工作流

任务工作流由工作流模型改进而来.定义任务工作流网为有向网 $N=(T,F)$,其中 T 为任务, F 为流关系,且需满足:

- 1) 存在惟一的起始任务 $T_{start} \in T$ 作为网路的起始点,其前驱为空;
- 2) 每个任务 $t \in T$ 都位于连接到 T_{start} 的一条通路,不存在孤立的任务;
- 3) 每一任务结束时必处于成功或失败 2 种状

态之一,2 种状态可分别指向不同的后继任务;

4) 任务间的依赖关系 F 可分为 2 类:发起一个尚未执行的任务,或终止一个正在执行的任务.

模型中只存在任务极其依赖的关系,虽然没有库所和标记这样的元素,但类似的功能已被整合到任务中.默认情况下任何任务必须等到所有前驱任务全部结束才能执行,相当于 AND_join 型;任务结束后其对应的所有后继流程也都会被执行,相当于 AND_split 型.其他的逻辑关系,如 OR_split 和 OR_join,将由特殊的任务来实现.不同之处在于区分出了“成功”和“失败”2 种互斥的结束状态,且流关系也包括“发起”和“终止”2 种.对任务首尾逻辑结构的处理不像 Petri 网那样显式地建模表示,而是用一个“工作流管理程序”来进行管理,这在编程实现时更加方便.

任务工作流有自己的图形语言,如图 4 所示,所有任务采用相同的图形样式.任务图形的头部可接纳多个前驱任务;图形的尾部分成 2 个区域,空白的一半表示执行成功,带叉的一半表示执行失败,从 2 个区域流出的箭头分别代表了 2 种状态下任务工作流网不同的路径.水下机器人任务流程中最为常见的成功或失败的分支流程在这里成为任务的内置属性.实箭头表示发起一个尚未运行的任务,虚箭头表示终止一个正在运行的任务.网络中的每个任务都有惟一的名称(称之为标号,以“#”开头),同类型的不同任务的名称也必须不重复.每个任务都有内置的超时值,默认情况下其值为无穷大,超时后任务会被判定为失败,网路流程转向其失败后的路由.

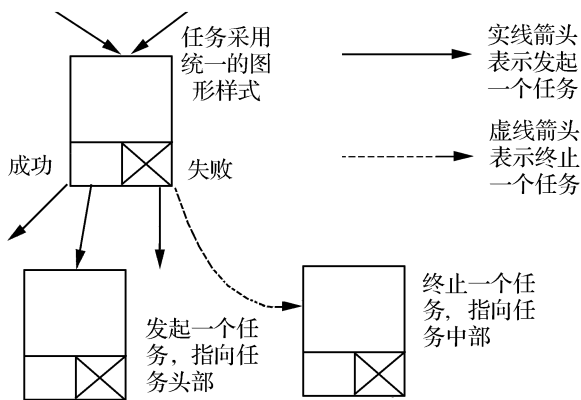


图 4 任务工作流图形描述方式

Fig.4 Graphic description of task workflow

每个任务都是不可分割的原子任务,要扩展某一任务,只能在相应的流程处插入子网.任务的具体内容由高级编程语言编写,任务获取环境信息、判断任务成功执行或失败的代码都在其实现代码中.任

务 workflow 模型只关注任务网络及流关系,并不规定任务的细节及具体实现方式。

OR_split 结构的实现有 2 种方式^[8]。其一如图 5 所示,在 T_4 、 T_5 前添加 T_2 、 T_3 2 个起着开关作用的判断任务,如果 T_4 执行的条件满足而 T_5 的条件不满足,则 T_2 任务成功而 T_3 任务失败,于是 T_4 执行而 T_5 不执行。另一种方式与具体的编程实现有关。在任务内部,可以访问并更改指向后继任务的流程。在上例中, T_1 内部本来保留有 T_2 、 T_3 的名称作为后继,如果在某种条件下删除其中 T_3 的名称,那么 T_1 完成后就只能执行 T_2 。这 2 种方式皆可在后继中任选数个任务来执行。

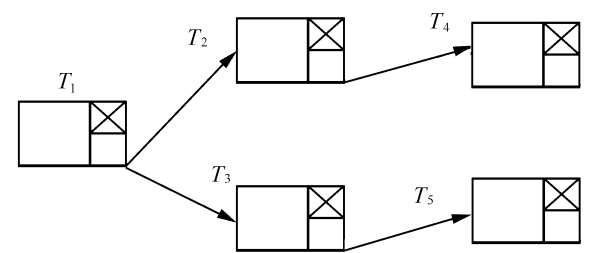


图 5 任务 workflow 中 OR_split 结构

Fig.5 OR_split structure in task workflow

图 6 中 T_1 是一个 OR_join 任务,这是任务 workflow 模型中惟一特殊的任务,注意 OR_join 后面的参数,第 1 个参数 2 表示 T_1 的前驱任务中任有 2 个结束后, T_1 就能以成功状态执行完毕。为了避免无止境的等待,OR_join 任务也有超时值。当第 1 个前驱任务结束时,OR_join 任务就被激活,处于等待状态。如果超过时限后,前驱条件仍未完全达成,此 OR_join 会被当作失败。

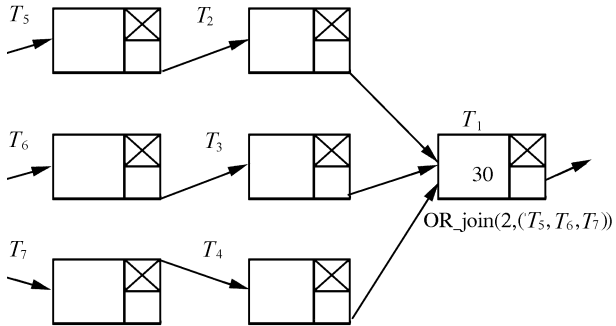


图 6 任务 workflow 中 OR_join 结构

Fig.6 OR_join structure in task workflow

为避免 OR_join 结构可能产生的逻辑错误,OR_join 任务在执行结束后会将其所有可能导致重复执行的前驱任务全部删除,以达到同步的效果。图

6 中,若 T_3 、 T_4 结束, T_1 成功,workflow 管理程序应将其直接的前驱 T_2 、 T_3 、 T_4 一并删除(删除即令运行中的任务停止,不考虑这些任务的任何后续影响)。

然而 OR_join 运行时的前驱条件并不总是如此简单,上例中,OR_join 任务的标记中还有另一串参数,显式指定给 workflow 管理程序并将其删除。于是 T_1 结束后会将图 6 中所有 6 个前驱任务全部删除(默认删除 T_2 、 T_3 、 T_4 ,显式删除 T_5 、 T_6 、 T_7)。这种方式强调对建模者意图的直接表述和利用,就模型本身而言并不严谨。然而在相关理论尚不完善之时,不失为一种简便有效的方案,相比更改或限制模型,以及用复杂的算法计算路径可达性的方法^[9-10],往往有着更好的实现效果。

循环结构的建模方式如图 7 所示,只需在其前方加入一个 OR_join 任务。

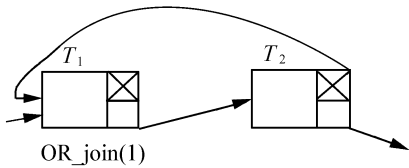


图 7 任务 workflow 循环结构

Fig.7 Circulation structure in task workflow

在精简了逻辑元素,整合了冗余的逻辑结构后,任务 workflow 能以更简洁的方式实现与其他任务模型相似的功能;而一些新特性的增加,如内置的成功或失败 2 种完成状态和超时值的应用,以及对 OR_join 结构逻辑问题的解决方案,不仅增加了模型的功能,也使得模型更加完善,更适合于水下机器人任务的建模。

4 任务 workflow 的建模流程

首先以图形化的方式对任务进行描述,如图 8 所示。对照图 8 可直接写出任务模型的文字描述:

```
#START { StartMission ( ) ; ; # ToTarget ; ; NULL ;
NULL ; NEVER }

#ToTarget { ToTarget ( ) ; #START ; #Join1 , #Record ; ; #Join2 ; ; }

#Join1 { OrJoin ( 1 ) ; # ToTarget , # Rearrange ;
#Work ; ; ; }

#Rearrange { Rearrange ( ) ; #Work ; #Join1 ; ; ; }

#Record { Record ( ) ; #ToTarget ; ; ; ; }

#Work { Work ( ) ; #Join1 ; #Join2 ; #Record ; #Rearrange ; ; }

#Join2 { OrJoin ( 1 ) ; #Work , #ToTarget ; #Return ;
; ; ; }
```



```
#Return { Return ( ) ; #Join2 ; #TakeBack ; ; #Move-  
Up ; ; 3 600 }  
#TakeBack { TakeBack ( ) ; #Return ; #Join3 ; ; ;  
; }  
#MoveUp { MoveUp ( ) ; #Return ; #Join3 ; ; ; }  
#Join3 { OrJoin ( 1 ) ; # MoveUp , # TakeBack ;  
#END ; ; ; }  
#END { EndMission ( ) ; #Join3 ; ; ; ; }
```

每一条描述语句的格式是相同的：
任务标号{任务类型及参数;前驱任务标号;成功执行后发起的任务标号;成功后终止的任务标号;失败后发起的任务标号;失败后终止的任务标号;超时值}。
描述语句中各项用分号隔开,若某一项任务标号为空,可以不写或写上“NULL”;若在超时值处为空或写上“NEVER”,则超时值为无穷。

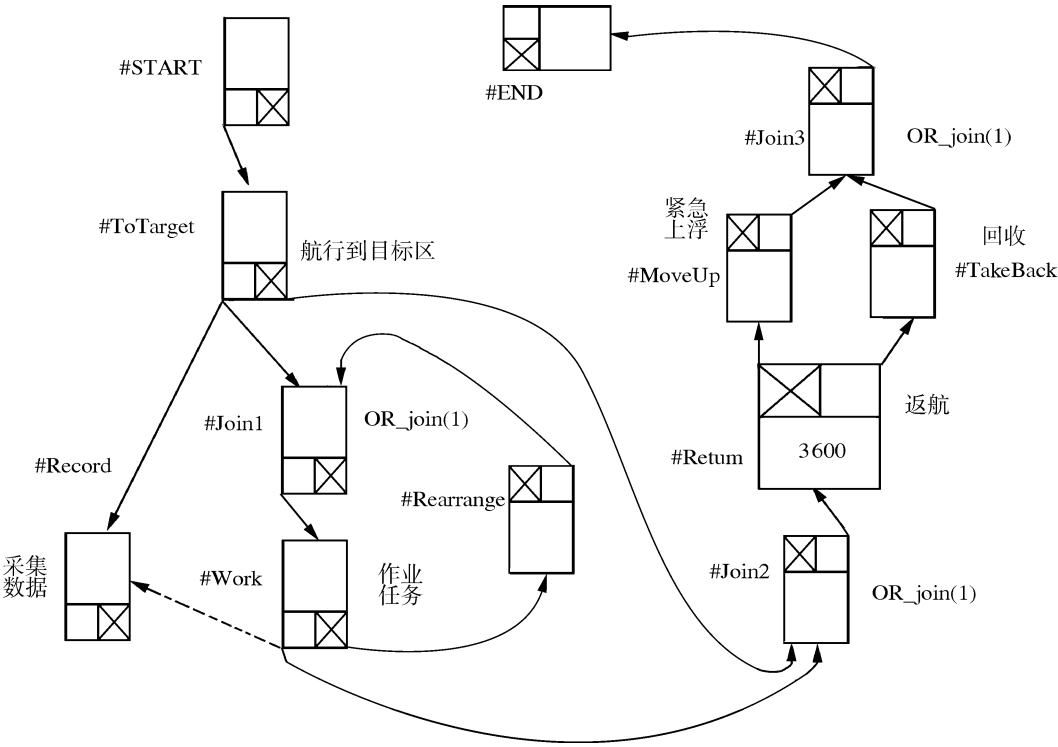


图 8 任务示例 (任务 workflow)
Fig.8 An example of AUV task flow (task workflow)

网路中任一表示流程的箭头,将既被发起它的前驱任务描述,又被接纳它的后继任务描述,任务间的依赖关系便由此衔接起来.对单个任务的简单叠加就得到了整个任务网.每一条语句就是对模型中一个任务及其外延的流关系的直接描述,故图中任务方框的个数与描述语句数相同.描述语句没有先后顺序可言,任务流程描述文件的格式是相当灵活的.规定这样的描述语言,是为了计算机能直接读取任务文本,自动按流程发起任务并对任务进行操作和管理.由于此描述语言格式规范,因此完全可以开发相应的应用软件,让建模者直接以图形方式建模,而由程序自动生成模型的文本描述.

在程序实现中,每个任务类对应一个 C++ 类,且每个任务类都继承统一的基类,在其中定义通用的数据类型和方法,包括对结束状态的判断,以及与外界通信的手段.如上例中, #Join1、#Join2、#Join3 都属于 OrJoin 类型,它们都会采用一份相同的代码来

实现.

5 任务管理程序的实现

作为实验室水下机器人模块化软件系统的一部分,基于上述原理与实现方式的水下机器人任务建模与管理系统的已经成功开发,并在仿真平台和陆地机器人实验中得到了验证.这里仅简述此系统的几个特点:

- 1)所有的任务类型需提前编写(用 C++ 语言)并编译完成.任务流程的描述由文本化的任务描述文件提供,系统读取文本中的任务类型名,自动生成对应的 C++ 类的实例并执行此任务.任务只有在需要时才动态生成,执行完成后销毁,以节约内存空间.这涉及到 C++ 中用字符串动态生成对象的技巧.
- 2)此系统提供任务流程更改的全套接口.无论任务位于文本描述文件中,还是正在内存中执行,都有标准的接口用于更改任务的流关系及类型,以便

在不更换描述文件的情况下更改任务流程。

3) 除起始任务、OR_join 任务等少数内置任务外,所有任务类型都由用户定义并编写,任务管理系统只是按照前文所述的方式自动读取并管理任务的执行流程。同一任务类型可以被多次使用,也可以给任务传递参数,使同一份任务代码完成不同的工作。由于任务可重用,故已经存在的任务越完善,建模的过程就越方便快捷。

4) 基于任务工作流的系统只关注任务间的流程关系,对系统的应用范围并不做规定。但相对而言,此模型更适合宏观抽象的任务流程建模,在机器人的自主智能尚不完善之时,将人类设计的抽象任务转换为机器可以理解的方式。

6 结束语

通过引入并改进工作流的相关理论,基于任务工作流的模型不仅建模简单(可完全依赖图形描述)、容易理解,更重要的是模型能直接转化为规范的文本描述供计算机读取,利于计算机编程实现。任务工作流对传统的工作流模型做了较多改进,在减少了模型组成元素的同时增强了模型的表述能力,并以合理的方式弥补了理论上的不足。在此基础上开发的水下机器人任务建模与管理系统的已有实际使用的实例,其性能稳定可靠。随着机器人任务流程进一步复杂化,专用的任务建模及管理系统必将发挥愈发重要的作用。

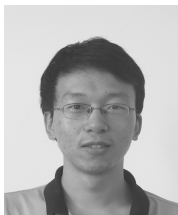
参考文献:

- [1] 袁崇义. Petri 网原理与应用[M]. 北京: 电子工业出版社, 2005.
- [2] 邹海, 边信黔, 熊华胜. AUV 控制系统规划层使命与任务协调方法研究[J]. 机器人, 2006, 28(6): 651-655.
ZOU Hai, BIAN Xinqian, XIONG Huasheng. Research on the mission and task coordination method of AUV control system at planning layer[J]. Robot, 2006, 28(6): 651-655.
- [3] 韩丽洁. 水下机器人任务规划技术研究[D]. 哈尔滨: 哈尔滨工程大学, 2010: 36-56.
- [4] VAN DER AALST W, VAN HEE K M. Workflow management: models, methods, and systems[M]. Cambridge, USA: The MIT Press, 2002.
- [5] 张亮, 姚淑珍. 基于 Petri 网的工作流模式研究[J]. 计算机集成制造系统, 2006, 12(1): 54-58.
ZHANG Liang, YAO Shuzhen. Research on workflow patterns based on Petri nets[J]. Computer Integrated Manufacturing Systems, 2006, 12(1): 54-58.
- [6] 王斌, 章云, 王晓红. 基于 Petri 网的工作流模式建模及应用[J]. 计算机工程与应用, 2008, 44(13): 238-241.

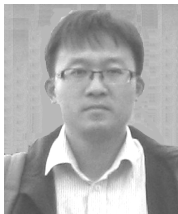
WANG Bin, ZHANG Yun, WANG Xiaohong. Modeling research and application of workflow patterns based on Petri nets[J]. Computer Engineering and Application, 2008, 44(13): 238-241.

- [7] VAN DER AALST W M P, TER HOFSTEDE A H M. YAWL: yet another workflow language[R]. Brisbane, Australia: Queensland University of Technology, 2003.
- [8] 陈荣辉, 陈新度, 陈新. 基 AND_OR 图深度优先搜索的工作流模型验证[J]. 机电工程技术, 2008, 37(6): 56-71.
CHEN Ronghui, CHEN Xindu, CHEN Xin. Workflow model verification based on AND_OR graph depth-first searching[J]. Mechanical & Electrical Engineering Technology, 2008, 37(6): 56-71.
- [9] 孙玲芳, 方锦烽, 汲铮, 等. YAWL 中 OR_join 任务使能判断的改进及形式化说明[J]. 江苏科技大学学报: 自然科学版, 2007, 21(5): 64-68.
SUN Lingfang, FANG Jinfeng, JI Zheng, et al. Improvement in enabling of OR-join in yet another workflow language[J]. Journal of Jiangsu University of Science and Technology: Natural Science Edition, 2007, 21(5): 64-68.
- [10] 曹国荣, 谭庆平, 吴浩, 等. 规范流网中 OR-join 任务的语义及使能判定算法[J]. 计算机科学与探索, 2010, 4(6): 542-551.
CAO Guorong, TAN Qingping, WU Hao, et al. Semantics and implementation of the OR-join in regular flow nets[J]. Journal of Frontiers of Computer Science and Technology, 2010, 4(6): 542-551.

作者简介:



张鸿皓, 男, 1988 年生, 硕士研究生, 主要研究方向为水下机器人控制软件系统设计。



姜大鹏, 男, 1981 年生, 讲师, 硕士生导师, 博士, 主要研究方向为多水下机器人协同作业技术、智能水下机器人技术。



庞永杰, 男, 1955 年生, 教授, 博士生导师, 主要研究方向为智能水下机器人技术、船舶操纵与控制技术。主持和参与省部级以上科研项目多项, 曾获国家科技进步二等奖 1 次, 国防科学技术一、二、三等奖各 1 次, 发表学术论文 50 余篇。