

文章编号: 1001-0920(2014)11-1927-08

DOI: 10.13195/j.kzyjc.2013.0952

## 基于 OPM/MAS 的钢铁企业多 agent 生产调度模型

王越, 苏宏业, 沈清泓, 朱理, 卢山

(浙江大学 a. 工业控制技术国家重点实验室, b. 智能系统与控制研究所, 杭州 310027)

**摘要:** 现有的面向 agent 建模方法在统一性、灵活性、交互能力、逻辑验证能力等方面存在很多不足, 阻碍了多 agent 系统的研究和设计. 为此, 引入对象过程/多智能体系统 (OPM/MAS) 的建模方法, 将系统的功能、结构和动态行为集成在统一的模型中, 用图形和自然语言共同表达复杂系统的抽象概念, 可进行逻辑验证, 并具有灵活性强、表意清晰等优势. 通过在钢铁企业生产调度中的应用, 验证了 OPM/MAS 在多 agent 等复杂系统建模中的有效性.

**关键词:** 对象过程/多智能体系统; 多主体; 钢铁生产调度

**中图分类号:** TP311

**文献标志码:** A

## Multi-agent production scheduling model of steel enterprise based on OPM/MAS

WANG Yue, SU Hong-ye, SHEN Qing-hong, ZHU Li, LU Shan

(a. state Key Laboratory of Industrial Control Technology, b. Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China. Correspondent: SU Hong-ye, E-mail: hysu@iipc.zju.edu.cn)

**Abstract:** Agent-oriented modeling tools have many deficiencies in uniformity, flexibility, interaction ability and logic verification capability, which hinder the research and design of multi-agent systems. An object-process methodology/multi-agent system(OPM/MAS) agent-oriented modeling method is introduced to express the abstractions of the complex system by using graphics and natural language. The function, structure and dynamic behavior of the system can be integrated in a refinery and unified model, and logic verification is available. OPM/MAS has advantages of flexibility and expressiveness. The application in the production scheduling of the steel industry shows the advantage of OPM/MAS as a modeling method for complex multi-agent systems.

**Key words:** object-process methodology/multi-agent system; multi-agent; production scheduling model of steel enterprise

### 0 引言

概念性模型是对实际系统的抽象描述, 对于系统的分析和设计具有重要的意义. 随着系统规模的不断扩大、复杂性的不断提高, 复杂系统的概念性建模越来越引起学者的关注<sup>[1]</sup>. Agent 理论的引入给复杂系统的分析和设计带来了新的解决途径. 目前, 使用比较广泛的面向 agent 建模方法有: Gaia、AUML (agent unified modeling language) 和 MaSE (multi-agent system engineering) 等<sup>[2]</sup>. Gaia 是基于知识工程的建模方法, 要求单个 agent 的能力以及 agent 之间的交互关系在系统运行过程中是静态的, 使得 agent 失去了自主性和反应性等特点<sup>[3]</sup>. AUML 是基于面向对象建模

方法 UML 的一种扩展. AUML 通过一系列模型, 从不同的视角来描述系统的各种静态结构和动态行为<sup>[4]</sup>. MaSE 是一种通用的、支持异构多主体系统的开发方法, 在系统分析阶段, 该方法为基于面向对象的建模理论<sup>[5]</sup>.

很多学者对面向 agent 的建模方法进行了改进, Farid 等<sup>[6]</sup>开发了可视化的建模工具软件 Maude, 可将 AUML 语言转化成可视化的动态模型, 弥补了 AUML 只能半动态地表达系统行为的不足. 张欧亚等<sup>[7]</sup>在模型表示法上对 AUML 进行了扩展, 分别用例图、agent 类、多种行为图来描述责任分配、系统静态结构和动态行为. Leila 等<sup>[8]</sup>将 AUML 和 Event-B 方法进行结合,

收稿日期: 2013-07-12; 修回日期: 2014-01-13.

基金项目: 国家科技支撑计划项目(2012BAF10B04); 高等学校学科创新引智计划项目(B07031).

作者简介: 王越(1985-), 女, 博士生, 从事制造执行系统、能源管理系统的研究; 苏宏业(1969-), 男, 教授, 博士生导师, 从事时滞系统、非线性系统、复杂工业过程建模和优化、制造运行管理与系统集成等研究.

规范了多 agent 系统交互协议. Innocent 等<sup>[9]</sup>基于 Petri nets 拓扑结构模型来描述各 agent 的空间分布及其动态行为. Regayeg<sup>[10]</sup>提出了一种基于 Z 语言和线性时序逻辑的新建模方法, 可以描述 agent 内部结构和交互协议. 虽然已有很多学者对面向 agent 的建模方法作出了改进, 但在统一性、灵活性、交互能力、表意能力、逻辑验证能力等方面仍具有很多不足, 阻碍了多 agent 系统理论的发展.

综上所述, 有必要研究一种新的概念性建模方法, 扩展原有建模方法的描述方式, 增强仿真模型描述的完整性, 使得仿真应用更符合客观实际. 本文引入了对象过程/多智能体系统 (OPM/MAS) 建模方法, OPM/MAS 是根据 agent 理论特点, 在 OPM 理论上进行的扩展. OPM 是综合面向对象、过程的建模方法, 可将系统的功能、结构和行为集成在一个统一的模型中, 并且由图形和自然语言来共同表达模型的含义<sup>[11]</sup>.

本文以钢铁企业为背景, 基于 OPM/MAS 建立统一的钢铁企业生产调度模型. 该模型既可描述生产调度中每个流程的静态结构、动态行为, 也可表达各流程之间的动态信息交互活动, 并且可通过动态仿真来验证生产调度模型的逻辑性, 为多模式、多路径、多设备下的生产调度提供分析平台和理论基础.

## 1 OPM/MAS 基本理论

OPM/MAS 是一种面向 agent 的概念性建模方法, 是基于 OPM 理论的扩展. OPM/MAS 秉承了 OPM 的建模优势, 并通过增加模型元素使其更适合于多 agent 复杂系统建模.

### 1.1 OPM 理论

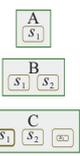
OPM 是一种综合面向对象和过程的建模方法, 适用于复杂系统建模, 由以色列理工学院 Dov Dori 教授提出, 并开发出支持 OPM 的软件平台 OPCAT<sup>[12]</sup>. OPM 用图形和自然语言来共同表达系统的抽象概念, 适合人类的思维方式, 可以应用在很多领域. 该理论具有 2 种描述方式、3 种实体、2 大链接类型, 通过实体和链接的不同组合表达系统的功能、静态结构和动态行为.

OPD 和 OPL 是 OPM 的两种描述方式. OPD 是一种图形语言, 通过提炼复杂系统的主要部分, 用图形来搭建模型, 并将重要过程或对象展开生成子模型来对系统作进一步的描述. 各个父子模型之间严格统一, 确保了系统中各个层面的细节不会被遗漏. OPL 是 OPD 的文字解释. OPL 是一种双目的性语言, 既面向人又面向机器, 是自然语言的子集, 并符合人类的

思维模式. OPL 可以辅助 OPD 使模型更易于理解, 增强了 OPM 的易理解性<sup>[13]</sup>.

对象、状态、过程是 OPM 的 3 种实体. 对象是存在的事物, 是组成 OPM 模型的基本单元, 可以是物理的也可以是信息的. 状态是对象所处的状况. 过程是改变对象的事件, 通过创建、消耗或改变对象的状态来影响对象<sup>[13]</sup>. 各实体图标如表 1 所示.

表 1 OPM 实体表

	对象	过程	状态
OPD			
	B is physical.	E is physical.	A is $s_1$ .
	C is environmental	F is environmental	B can be $s_1$ or $s_2$ .
OPL	and physical.	and physical.	C can be $s_1$ , $s_2$ , or $s_3$ .
			$s_1$ is initial.
			$s_3$ is final.
解释	对象是存在的事物, 用矩形表示.	过程是改变对象的事件, 用椭圆表示.	状态是对象所处的一种状况或可假设的具 体值. 状态伴随对象而存在, 具有初始状态和最终状态.

OPM 的链接类型分为结构链接和时序链接, 实体之间通过这两种链接进行组合, 可以表达系统各单元之间静态和时序上的关系. 时序链接应用于过程与对象或过程与对象的状态之间来表达系统的行为. 系统的行为有 3 种方式: 1) 过程产生、消耗对象或改变对象状态; 2) 对象触发过程发生, 但不被过程所转化; 3) 若条件满足, 对象可以触发带有过程的事件. 与这 3 种行为对应的链接分别为转化链接、使能链接和事件链接.

结构链接用于描述实体间的结构关系, 具有 4 种类型: 整体-部分结构、展示-属性结构、普遍-特殊结构和类别-特例结构. 整体-部分结构展示了某事物整体与部分之间的结构关系; 展示-属性结构展示了某事物与其属性之间的结构关系; 普遍-特殊结构展示了普遍事物与特殊类别事物间的结构关系; 类别-特例结构展示了一类事物与其中一个具体事例之间的结构关系.

OPM 作为新一代既面向对象又面向过程的概念性建模方法, 给概念性建模提供了新的方向, 广泛应用于计算机集成制造、系统开发与设计、企业资源计划、生物工程等众多领域, 引起了很多学者的关注. 文献 [14] 将 OPM 应用在企业资源计划 (ERP) 当中, 建立了详细、普适的 ERP 模型, 解决了以往的 ERP 模型

通用性差的问题. 文献[15]用 OPM 替代面向对象的建模方法, 将其应用到软件体系结构中, 解决了面向对象建模方法在系统动态行为方面描述不足的缺点. 文献[16]将 OPM 应用在网络服务组合中, 准确、可视化地呈现了组合过程和结果. 文献[17-18]将 OPM 引入软件重用当中, 为现有软件的二次开发提供了有力工具, 并节省了大量资源. 文献[19]将 OPM 应用于生物过程的建模中, 对信使核糖核酸的全生命周期过程建立了统一的静、动态模型, 形象、准确地描述了整个信使核糖核酸的转换过程.

## 1.2 OPM/MAS

目前, 面向 agent 的建模方法繁多, 但是每种方法都有自己独特的建模理论, 没有统一的模型元素和模型规范. OPM/MAS 在 OPM 的基础上引入了新的模型元素, 扩展了 OPM 的建模能力<sup>[20]</sup>. 以下是引入的模型元素:

agent 为具有控制问题求解能力的计算单元;

environment 为 agent 所处的逻辑环境;

message 为 agent 之间进行交互的平台;

task 为需要由 agent 执行的带有约束的任务;

role 为 task 的抽象描述, 一般指某 agent 所具有的功能;

protocol 为交互协议;

organization 为由多个 agent 所组成的 agent 群, 通过多个 agent 的相互协作来完成共同目标;

mobilizing 为支持 agent 进行角色转换的过程, 具有类型、地点、优先级、结果等属性.

引入新的模型元素可对多 agent 系统进行针对性的描述. 其中: role 用来表达 agent 及 agent 群的功能; agent、environment、organization 用来展示多 agent 系统的整体结构; message、task、protocol、mobilizing 用来描述 agent 的动态行为及各 agent 之间的交互方式, 使模型更加清晰、规范.

## 2 OPM/MAS 在钢铁企业多 agent 生产调度中应用

### 2.1 钢铁制造过程描述

钢铁生产是一种连续与离散相混合的生产过程, 工序繁多, 设备复杂. 炼钢、连铸和热轧是钢铁生产流程的 3 大关键工序. 铁水和废钢等原料在转炉内经过吹氧、脱硫、脱磷等工艺产生符合要求的钢水, 钢水经过精炼后通过连铸工序生产出不同规格的中间品板坯, 并存入中间品库存中等待轧制. 根据订单中产品的需求, 轧制过程将板坯加工成不同规格和用途的成品.

### 2.2 钢铁生产调度的目标和任务

钢铁生产过程的每道工序都有自己的生产条件和约束, 为了建立“炼钢-连铸-热轧”的一体化调度模型, 有必要充分考虑每道工序的约束和优化目标. “炼钢-连铸-热轧”集成调度的任务是确定满足炼钢、连铸、热轧每道工序生产约束的炉次顺序、浇次顺序和轧制顺序, 使产品满足客户的要求, 并尽可能地降低成本<sup>[21]</sup>. 集成调度需要考虑的问题主要有以下几个方面:

1) 工作订单的组合问题: 哪些订单应组合在一个生产单元中, 哪些订单可以由板坯的匹配来完成生产任务;

2) 每个工序任务的内部排序: 根据工序的约束和生产材料的规格来确定相应的生产序列;

3) 工序之间的衔接问题: 如何尽可能地减少工序之间的加工等待时间.

### 2.3 利用多 agent 技术的必要性

钢铁企业“炼钢-连铸-热轧”一体化集成调度问题, 是一个复杂的带有巨大能量流和物流的多级调度问题. 就单一的工序而言, 炼钢、连铸、热轧的调度问题已经是 NP-hard 问题, 而三者结合的一体化问题则更为复杂, 涉及到许多控制变量, 解空间指数级增长, 造成了求解的极大困难, 甚至可能出现无可行解的情况. 多 agent 系统是由多个 agent 有机组成, 是分布式人工智能的重要课题, 具备分散控制和并行处理的能力, 适合于分布式计算和不确定问题求解<sup>[22]</sup>. 因此利用多 agent 技术可以有效解决钢铁企业“炼钢-连铸-热轧”一体化集成调度问题.

### 2.4 多 agent 系统结构

Agent 是具有控制问题求解能力的计算单元, 它可以是机器人、专家系统也可以是单元模块. 在功能上, agent 可以与外部环境交互, 然后对得到的信息进行加工处理, 再反作用于环境. 多 agent 系统(MAS)是由多个 agent 有机组成通过知识交换、控制信息传递等完成对各 agent 间的协调, 实现各 agent 的并行计算<sup>[23]</sup>.

本文将钢铁生产过程的主要任务定义成 agent 或 agent 群. 整个调度系统由订单 agent、炼钢 agent 群、连铸 agent 群、热轧 agent 以及板坯库 agent 组成. 每组 agent 都有自己的任务和目标, 最后通过 agent 之间的协调完成一体化集成调度. 其中各组 agent 之间通过“黑板式”通讯方式进行交互. 对于炼钢 agent 群和连铸 agent 群, 每组群内之间的 agent 组成联盟式的多 agent 系统结构. 在这种结构体系下, agent 之间不

需要知道其他所有 agent 的详细信息, 灵活性强, 并且可以有效地降低系统通讯的复杂性.

### 2.5 基于 OPM/MAS 的多 agent 生产调度模型

对于多 agent 系统, 每个 agent 是既具有内部结构关系又具有动态行为的智能体. 利用 OPM/MAS 的建模优势对钢铁企业多 agent 生产调度系统建立静态动态结合的统一模型是合理和可行的. 图 1 和图 2 是基于 OPM/MAS 炼钢企业生产调度的概念性模型, 旨在对 OPM/MAS 的模型描述能力进行展示. 图 1 是对多 agent 生产调度模型的整体描述, 将生产调度系统分成多个智能主体: 订单 agent、炼钢 organization、连铸

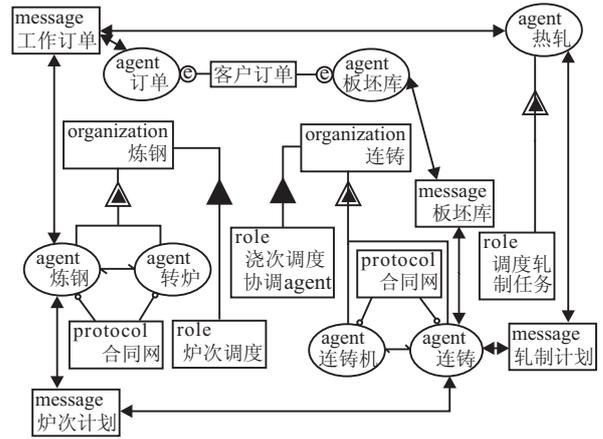


图 1 多 agent 生产调度总模型

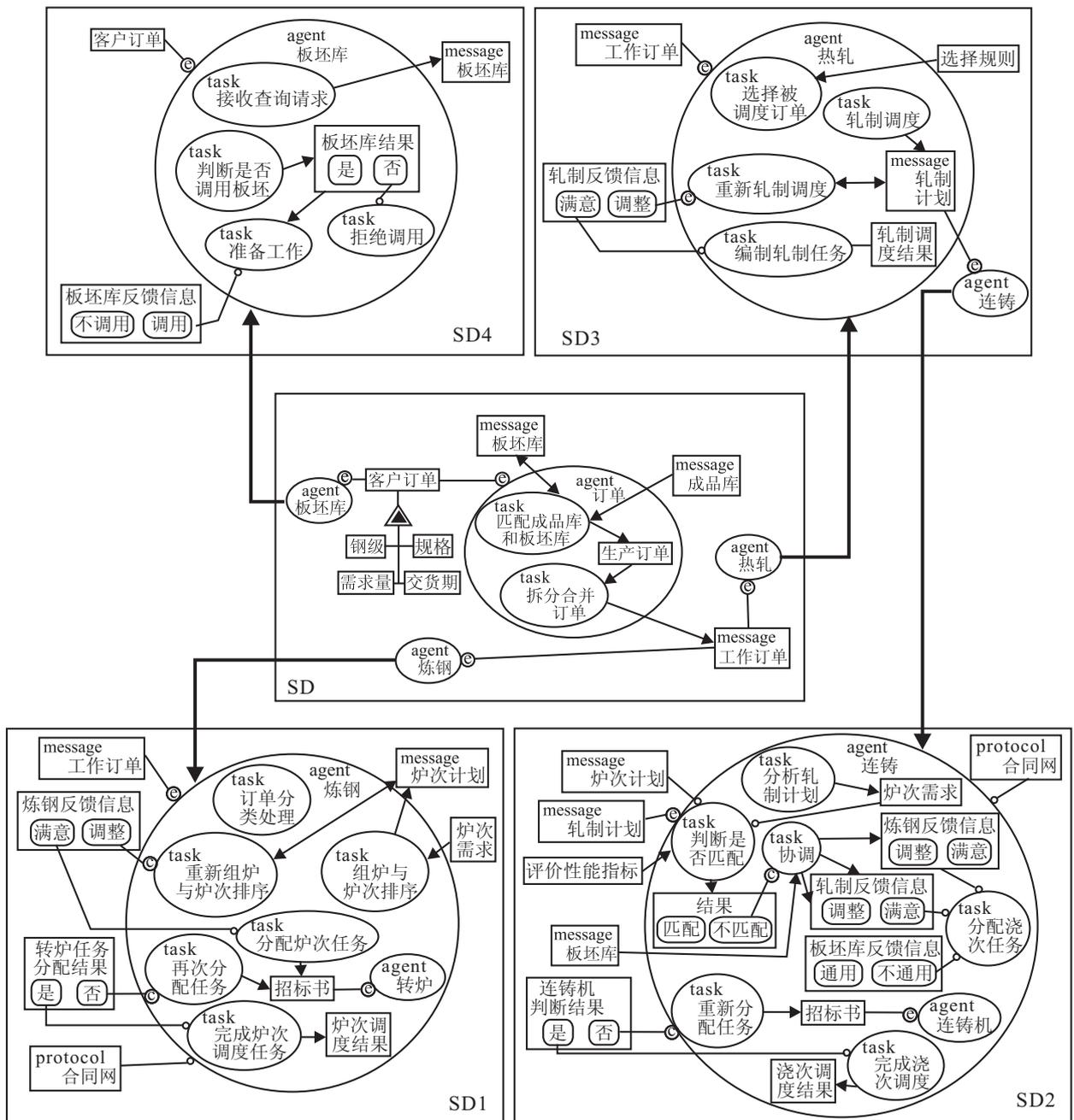


图 2 多 agent 一体化集成调度模型

organization、热轧 agent 和板坯库 agent, 每个智能主体具有不同的职责. 其中连铸和炼钢 organization 是具有多个 agent 的 agent 群. 图 1 中, 炉次计划、轧制计划、板坯库和工作订单 message 作为“黑板”, 实现各组 agent 间的动态交互.

客户订单作为系统的输入, 通过设备链接(一种时序链接)与订单 agent 和板坯库 agent 相连, 表达两者对客户订单进行处理的动态行为. 各 agent 与 message 之间通过作用链接(一种时序链接)相连, 表达各 agent 之间交互的动态行为. 炼钢 agent 与转炉 agent 之间由双向的结构链接相连, 表达两者之间的交互关系. 图 1 是对系统主要部分的概括性描述, OPM/MAS 可以对图中的每一个实体进行展开, 详细地描述各自的功能、结构和行为. 图 2 中各子图分别是对各个 agent 展开后生成的子模型. 在 OPM/MAS 中, 子模型与上级模型之间是父子关系, 并保持严格的统一性, 共同描述完整、统一的同一个模型.

图 2 中 SD(system diagram) 是对订单 agent 进行扩展后生成的子模型. 订单 agent 的任务是按照客户订单的钢级、规格、需求量、交货期等属性对订单进行预处理, 匹配成品库和板坯库; 对于不能由成品或板坯匹配来完成的生产订单, 按照工艺过程进行工序拆分形成工作订单. 图 2 中 SD 详细地描述了客户订单的静态结构, 展示了其内部属性. Message 成品库与 task“匹配成品库和板坯库”通过转化链接(一种时序链接)相连, 表达对象被过程转化的行为. Message 工作订单通过设备链接同时触发炼钢 agent 和热轧 agent.

图 2 中 SD1 是对炼钢 agent 进行展开后生成的子模型. 炼钢 agent 的职责是编排整个工序的钢水出炉序列; 安排每台转炉的炉次计划. 其具体任务为: 根据工作订单的规格和炼钢生产的约束进行订单分类处理; 完成炼钢工序的组炉和炉次排序任务; 根据连铸 agent 产生的 object“炼钢反馈信息”调整组炉和炉次计划; 最后完成多个转炉的炼钢任务分配, 通过炼钢 agent 的协调, 达成一致, 完成炉次的调度任务.

炼钢 agent 群由 1 个炼钢 agent 和多个转炉 agent 组成, 其中炼钢 agent 是中介 agent, 完成群内部各转炉 agent 间的协调工作以及群外部与其他 agent 的交互任务. 炼钢 agent 接受来自订单 agent 输出的信息“message 工作订单”, 对订单进行分类处理, 并进行组炉. 被分配到同一炉次的订单应满足如下 3 个条件: 1) 订单的钢级、宽度、厚度应一致; 2) 交货期接近; 3)  $95\% \text{炉容量} \leq \text{同炉次内板坯总重量} \leq 100\% \text{炉容量}$ .

炼钢 agent 群内部建立了炼钢 agent 作为招标者、

多个转炉 agent 作为投标者的合同网模型. 炼钢 agent 将每个炉次的任务作为一个招标书, 每个招标书包括炼钢的钢级、规格、炉次钢水的重量、完成时间等, 然后对招标书进行招标. 转炉 agent 接收到招标书之后, 评估自身是否具有投标的能力, 如有则发送标书到炼钢 agent, 并提供自己的执行能力. 炼钢 agent 在收到投标后, 对标书进行评价, 并选择评价结果最优的投标书, 评定其为中标, 即将指定的炉次分配给中标的转炉. 在评定投标书的过程中可能会出现特殊的情况, 例如没有收到投标书. 此时, 需要炼钢 agent 重新进行炉次的调度.

SD1 中, object“炼钢反馈信息”的 state“调整”与 task“重新组炉及炉次排序”通过条件链接(一种时序链接)相连表示, 若 object“炼钢反馈信息”进入 state“调整”, 则执行 task“重新组炉及炉次排序”; 否则, 将跳过 task“重新组炉及炉次排序”, 继续执行下面任务.

Object“炼钢反馈信息”的 state“满意”与 task“分配炉次任务”通过使能链接(一种时序链接)相连表示, 若 object“炼钢反馈信息”进入 state“满意”则执行 task“分配炉次任务”; 否则, 一直等待, 直到进入 state“满意”才继续执行下一任务. OPM/MAS 除了用时序链接描述系统时序上的行为以外, 还按照过程的位置高度, 从上至下地执行每个过程. 因此 task“订单分类处理”之后将执行下面的 task“组炉与炉次排序”.

图 2 中 SD2 是将连铸 agent 进行展开后生成的子模型. 连铸 agent 除了负责连铸工序浇次的调度任务之外, 还扮演着与其他主要 agent (炼钢 agent、热轧 agent、板坯库 agent 等) 进行协调交互的角色, 在整个多 agent 生产调度系统中占枢纽地位. 主要任务包括: 解决浇次计划与轧制计划之间的冲突, 权衡并协调轧制热送率与无委材之间的冲突. 此处不再赘述.

如 SD2 所示, 首先由热轧 agent 产生的 message“轧制计划”来触发连铸 agent. 首先由热轧 agent 产生的 message“轧制计划”来触发连铸 agent. 然后连铸 agent 执行 task“分析轧制计划”, 并产生 object“炉次需求”. 当 object“炉次需求”与由炼钢 agent 产生的 message“炉次计划”同时存在时, Task“判断是否匹配”被激活. 若判断结果不匹配, 则执行 task“协调”; 否则跳过 task“协调”执行下一 task“分配浇次任务”. task“重新分配任务”由来自连铸机 agent 产生的 object“连铸机判断结果”的 state“否”触发, 直到各 agent 达成一致, 完成浇次调度任务.

图 2 中 SD3 是对热轧 agent 进行扩展后生成的子模型. 热轧 agent 的任务是在订单候选池中选择被调度的订单, 对轧制单元进行调度排序; 根据由连铸

agent 产生的 object“轧制反馈信息”，重新选择订单，并重新进行调度计算，直到各连铸机 agent 达成一致，完成轧制调度任务。

图 2 中 SD4 是对板坯库 agent 扩展后生成的子模型。板坯库 agent 的主要任务是：收到订单 agent 的查询请求后，生成匹配信息；根据由连铸 agent 产生的 object“板坯库反馈信息”判断是否调用板坯，并对请求作出应答。

### 3 基于 OPM/MAS 的多 agent 生产调度模型分析

由于面向 agent 建模方法的多样性，不同方法之间在建模概念、建模过程和建模语言等方面存在很多差异。学者开始关注对面向 agent 建模方法进行系统地分析、比较和评价，从而为多 agent 系统的设计选择合适的建模方法提供依据，发现不同方法的优缺点，并进一步为面向 agent 系统建模方法实现标准化奠定基础。本文结合钢铁企业多 agent 生产调度模型，在统一性、逻辑验证能力、交互能力、灵活性和表意能力等方面，对 OPM/MAS、Gaia、MaSE、AUML 这几种应用比较广泛的面向 agent 建模方法进行对比分析，并验证了 OPM/MAS 在以上方面对多 agent 系统建模的优越性。

#### 3.1 统一性

OPM/MAS 可将多 agent 系统的功能、结构和行为集成在一个统一的模型中，既保证系统的完整性，又展现系统的诸多方面。在图 1 中，OPM/MAS 不但描述了多 agent 生产调度系统由 5 组 agent 构成的静态结构关系，而且表达了各 agent 之间的交互方式，以及

各 agent 的功能作用。其中“炼钢 organization”由结构链接与“炼钢 agent”和“转炉 agent”相连，表达其由“炼钢 agent”和“转炉 agent”组成。“炼钢 organization”的功能由“role 炉次调度”来展示，其动态行为则通过图 2 中 SD1 炼钢 agent 子模型来描述。图 1 和图 2 统一地呈现了钢铁企业多 agent 生产调度系统的功能、结构和行为。

与 OPM/MAS 相比，AUML 割裂了系统的整体性和统一性，其通过例图、类图、时序图和状态图分别表达系统需求、静态结构、动态行为和反应能力。对于多 agent 系统而言，每个 agent 既具有动态行为又具有内部结构，用统一的建模方法对其进行描述更为合适。

#### 3.2 逻辑验证能力

OPM/MAS 的逻辑验证能力主要体现在消除模型的不一致性，验证模型的逻辑结构是否符合实际，是否准确地表达了真实的系统。模型的不一致性是指，同一个对象或过程具有不同名称或表示类型的情况，这是一种模型元素指代的不一致。OPM/MAS 的软件环境 OPCAT 可以消除模型不一致性，并在仿真过程中提供调试日志和生命周期图。调试日志提示逻辑错误出现的位置。生命周期图可以呈现模型中所有对象和过程在调试过程中的状态，有利于建模者深入分析造成逻辑错误的原因。当模型逻辑调试成功之后，编译模型，OPD 中将出现一个红色圆点，沿着模型的逻辑顺序遍历整个模型结构，并在各个父子模型之间切换，使建模者可以直观、清晰地分析模型是否符合实际系统或设计思路。

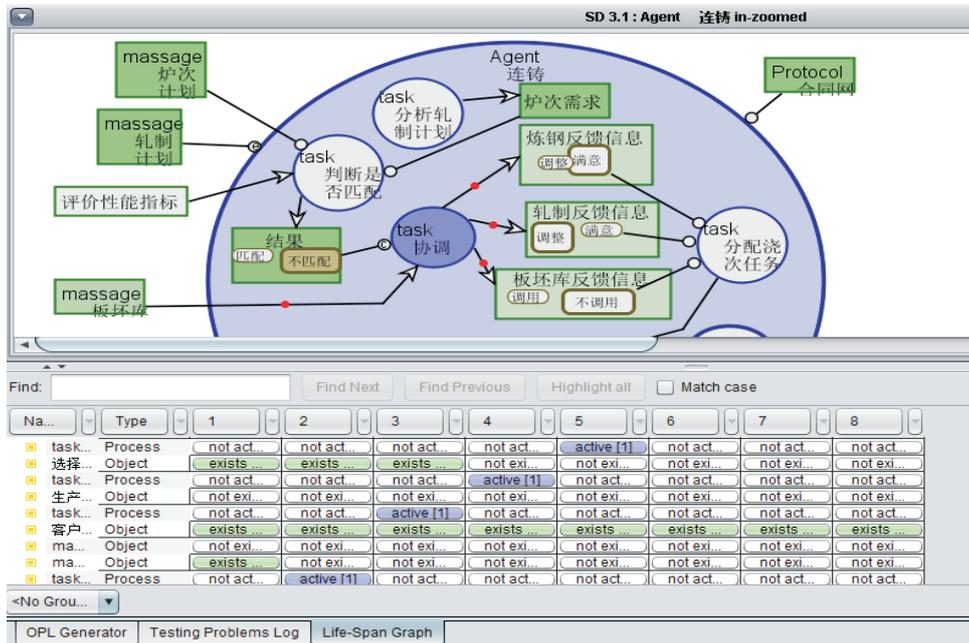


图 3 OPM/MAS 动态仿真图

由于连铸 agent 起到协调上下游 agent 的作用, 所以内部逻辑关系复杂, 给逻辑验证带来一定难度. OPM/MAS 能够清晰地呈现连铸 agent 行为的各个可行路径, 为模型的准确性和模型与实际系统的吻合性提供了保障. 连铸 agent 子模型的动态仿真如图 3 所示. 其他 agent 的建模语言(如 MaSE、AUML)都是基于面向对象的建模方法, 通过不同的类图来描述系统, 需要建模者通过抽象思维来整合系统不同方面的信息, 给模型的建立带来了困难. OPM/MAS 的这种逻辑验证功能, 为多 agent 系统建模提供了方便.

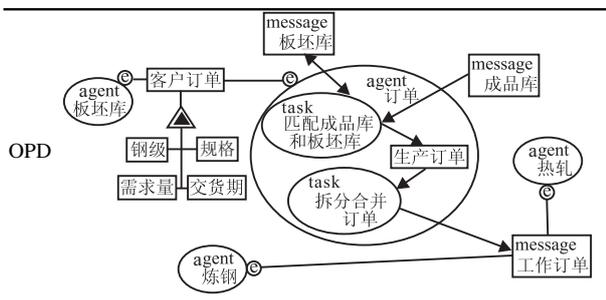
### 3.3 交互能力

OPM/MAS 通过扩展的模型元素 mobilizing 来实现 agent 的角色转换. 在系统运行中, 允许 agent 改变交互关系, 具有较强的交互能力. Gaia 要求单个 agent 的能力以及 agent 之间的交互关系在系统运行过程中是静态的, 使得 agent 失去了自主性和反应性等特点. MaSE 要求 agent 的交互是单对单, 而不是多点传送, 更加限制了 agent 的自主性.

### 3.4 灵活性

OPM/MAS 允许建模者增加 OPM 的扩展元素, 以适应面向 agent 系统的发展. 如图 1 中的 message、agent、role、organization、task、protocol 可以根据系统的需要进行调整. 而 Gaia、AUML 和 MaSE 具有严格的元模型定义, 不允许建模者对其进行修改, 缺乏灵活性.

表 2 OPM/MAS 模型中 OPD 与 OPL 对比表



OPD	message 工作订单 triggers agent 炼钢 and agent 热轧. 客户订单 exhibits 钢级, 规格, 需求量, and 交货期. 客户订单 triggers agent 板坯库 and agent 订单. agent 热轧 requires message 工作订单. agent 板坯库 requires 客户订单. agent 炼钢 requires message 工作订单. agent 订单 exhibits 生产订单.
OPL	agent 订单 consists of task 匹配成品库和板坯库 and task 拆分合并订单. agent 订单 requires 客户订单. agent 订单 zooms into task 匹配成品库和板坯库 and task 拆分合并订单, as well as 生产订单. task 匹配成品库和板坯库 affects message 板坯库. task 匹配成品库和板坯库 consumes message 成品库. task 匹配成品库和板坯库 yields 生产订单. task 拆分合并订单 consumes 生产订单. task 拆分合并订单 yields message 工作订单.

### 3.5 表意能力

OPM 由图形和自然语言来共同表达模型的含义, 更适合人类的思维方式. 表 2 以订单 agent 为例, 对比分析了订单 agent 子模型中 OPD 和 OPL 的联系和区别. OPD 是一种图形语言, 通过对复杂系统进行提炼, 用图形来搭建模型. OPL 是自然语言的子集, 使用人类最熟悉的表达方式描述模型. OPL 可以辅助 OPD 使模型更易于理解, 增强了 OPM 的易理解性.

## 4 结 论

面向 agent 的建模方法存在很多不足, 而且各种方法之间通用性差, 导致多 agent 系统的应用并不十分广泛. 目前面向 agent 建模方法的研究现状与 20 世纪 80 年代面向对象建模方法的研究状况极为相似, 需要一种统一的、能够被大多数学者认可的标准化方法. 本文引入了一种基于 OPM 的面向 agent 建模方法 (OPM/MAS), 可以对多 agent 系统建立统一的静动态模型, 具有交互能力强、灵活性强、表意清晰等优势. 以钢铁企业多 agent 生产调度系统为对象, 建立了基于 OPM/MAS 的多 agent 模型, 并进行了逻辑验证, 体现了 OPM/MAS 在多 agent 等复杂系统建模中的优越性, 为钢铁行业多 agent 系统的设计和开发提供了一定的理论依据.

### 参考文献(References)

- [1] 刘洁, 柏彦奇, 孙海涛. 概念模型建模方法研究[J]. 长春理工大学学报: 自然科学版, 2007, 30(3): 126-130.  
(Liu J, Bai Y Q, Sun H T. Research on conceptual modeling[J]. J of Changchun University of Science and Technology: Natural Science Edition, 2007, 30(3): 126-130.)
- [2] 李瑛, 毕义明, 刘顺成. 一种面向 agent 的概念化建模方法[J]. 指挥控制与仿真, 2009, 31(4): 15-19.  
(Li Y, Bi Y M, Liu S C. An agent-oriented approach for conceptual model[J]. Command Control & Simulation, 2009, 31(4): 15-19.)
- [3] 薛德黔. 软件工程[M]. 北京: 科学出版社, 2005: 207-208.  
(Xue D Q. Software engineering[M]. Beijing: Science Press, 2005: 207-208.)
- [4] 蒋美云. 基于 UML 的面向 agent 建模[J]. 软件导刊, 2011, 10(11): 3-5.  
(Jiang M Y. Agent-oriented modeling based on UML[J]. Software Guide, 2011, 10(11): 3-5.)
- [5] 毛新军. 面向主体的软件开发[M]. 北京: 清华大学出版社, 2005: 152-155.  
(Mao X J. Agent-oriented software development[M]. Beijing: Tsinghua University Press, 2005: 152-155.)

- [6] Mokhati F, Sahraoui B, Bouzaher S. A tool for specifying and validating agents' interaction protocols: From agent UML to maude[J]. *J of Object Technology*, 2010, 9(3): 59-77.
- [7] 张欧亚, 佟明安. 面向 agent 的巡航导弹武器控制系统分析[J]. *系统工程与电子技术*, 2007, 29(2): 237-242. (Zhang O Y, Tong M A. Agent-oriented analysis of cruise missile weapon control systems[J]. *System Engineering and Electronics*, 2007, 2(29): 237-243.)
- [8] Jemni Ben Ayed L, Siala F. Specification and verification of multi-agent systems interaction protocols using a combination of AUML and event B[C]. *Interactive Systems -Design, Specification, and Verification*. Heidelberg: Springer, 2008: 102-107.
- [9] Bakam I, Kordon F, Le Page C, et al. Formalization of a spatialized multiagent model using coloured petri nets for the study of an hunting management system[C]. *Interactive Systems-design, Specification, and Verification*. Heidelberg: Springer, 2001: 123-132.
- [10] Regayeg A, Kacem A H, Jmaiel M. Specification and verification of multiagent applications using temporal Z[C]. *Intelligent agent Technology Conference*. Beijing, 2004: 260-266.
- [11] Dori D. Object-process methodology — A holistic systems paradigm[M]. Heidelberg: Springer, 2002: 4-5.
- [12] Dori D. Rapid conceptual design for complex systems[EB/OL]. [2013-05-01]. <http://www.opcat.com/>.
- [13] Sturm A, Dori D, Shehory O. An object-process-based modeling language for multiagent systems[J]. *IEEE Systems, Man, and Cybernetics Society*, 2010, 40(2): 227-241.
- [14] Soffer P, Golany B, Dori D. ERP modeling: A comprehensive approach[J]. *Information Systems*, 2003, 28(6): 673-690.
- [15] Liu H, Gluch D P. Conceptual modeling with the object-process methodology in software architecture[J]. *J of Computing Sciences in Colleges*, 2004, 19(3): 10-21.
- [16] Liu Y, Liu W, Jiang C. Object-process diagrams as explicit graphic tool for WEB service composition[J]. *J of Integrated Design and Process Science*, 2004, 8(1): 113-127.
- [17] Iris R B, Dori D, Shmuel K. Reusing semi-specified behavior models in systems analysis and design[J]. *Software and Systems Modeling*, 2009, 8(2): 221-234.
- [18] Galia S, Iris R B, Dori D. Modeling design patterns for semi-automatic reuse in system design[J]. *J of Database Management*, 2010, 21(1): 29-57.
- [19] Dori D, Choder M. Conceptual modeling in systems biology fosters empirical findings: The mRNA lifecycle[J]. *PLoS ONE*, 2007, 2(9): 1-14.
- [20] Arnon Sturm, Dov Dori, Onn Shehory. Single-model method for specifying multi-agent systems[C]. *Proc of the 2nd International Joint Conference on Autonomous Agents and Multi-agent Systems*. New York, 2003: 121-128.
- [21] 纪日耿. 基于 UML 的钢铁企业 APS 订单计划系统建模[D]. 杭州: 浙江大学控制科学与工程学院, 2010. (Ji R G. APS order schedule system modeling based on uml for steel enterprise[D]. Hangzhou: Zhejiang University, Control Science and Engineering Institute, 2010.)
- [22] Pavel T. Multi-agent system design and integration via agent development environment[J]. *Engineering Applications of Artificial Intelligence*, 2012, 25(4): 846-852.
- [23] Lane D M, Mcfadzean A G. Distributed problem solving and real-time mechanisms in robot architectures[J]. *Engineering Applications of Artificial Intelligence*, 1994, 7(2): 105-117.

(责任编辑: 孙艺红)