

基于语义 Web Service 的模型自动组合综述

黄辉¹ 陈学广¹ 王志武²

(华中科技大学控制科学与工程系 武汉 430074)¹ (河南工程学院计算机科学与工程系 郑州 451191)²

摘要 回顾了 DSS 模型管理功能中模型组合概念的提出、发展和实现。对当前分布式模型管理以及模型组合的研究现状进行了综述,比较了主流模型自动组合设计方法各自的优缺点。由于 Web Service 技术的出现,以及语义 Web、人工智能规划等领域的创新应用,使得模型的自动组合有了实现的可能,但同时也面临难题:目前的分布式模型组合主要借鉴 Web Service 的自动组合技术,通过将模型封装为模型服务来把模型组合的问题转化为 Web Service 组合。然而模型组合有其自身的特点,例如定性和定量模型不能直接组合,使得传统的 Web Service 自动组合方法不完全适用。

关键词 分布式 DSS, 语义 Web service, 模型自动组合

中图分类号 TP391 **文献标识码** A

Survey of Automatic Model Composition Based on Semantic Web Service

HUANG Hui¹ CHEN Xue-guang¹ WANG Zhi-wu²

(Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)¹

(Department of Computer Science and Engineering, Henan Institute of Engineering, Zhengzhou 451191, China)²

Abstract This paper reviewed the proposition, development and implementation of the model composition function in decision support systems (DSS). A survey of distributed model management and model composition was given, and multiple model composition methods were compared. Due to the emergence of Web Service technology, and the innovative application of both of the semantic Web and AI planning fields, the automatic composition of DSS models is possible. But some problems also exist. The distributed model composition methods mainly learn from the automatic Web service composition, and models are encapsulated into Web services. The problem of model composition is converted to Web service composition. However, model composition has its own characteristics that the traditional automatic Web service composition methods are not fully applicable, for example the direct composition of qualitative and quantitative models.

Keywords Distributed decision support systems, Semantic Web service, Automatic model composition

1 引言

模型是对客观事物的一种抽象描述,在人们制定决策的过程中扮演着重要的角色。模型库系统是一个软件系统,它提供了工具和环境来开发、存储和操作模型,数据和与复杂决策问题相关的解决方法。模型库系统是决策支持系统(Decision Support Systems, DSS)的核心部件,通过模型或者模型的组合来辅助决策是 DSS 的中心思想。模型库系统由模型库(Model Base, MB)和模型库管理系统(Model Base Management System, MBMS)组成。

MBMS 的概念最初由 Sprague 和 Carlson^[1]提出,并得到学术界的一致认可和重视。模型管理,是为了给组织的决策制定提供广泛的支持^[2,3],并提供对组织模型的集中化管理^[4]。模型管理的主要功能包括:模型表示、模型的存储、

模型的查询和维护、模型选择(model selection)、模型组合(model composition)、模型集成(model integration)、模型和数据的链接(linking models and data for execution)^[5]等等。对这些功能的研究和实现,在 20 世纪 80—90 年代达到高峰,并取得了丰富的成果。但由于当时的计算技术和网络条件的限制,这些研究大多是在传统的集中式处理环境下进行的。

互联网的发展改变了信息系统的设计、发展和实现,也诞生了一种关于系统化的制定决策的新想法,这就是决策支持技术的分布式应用^[6],利用分布式技术设计和实现的决策支持系统叫做分布式决策支持系统(Distributed Decision Support Systems, DDSS)。

在 DDSS 环境下,决策模型和数据资源分布在不同的地理位置,它们由计算机网络相连并可能使用不同的操作平台。DDSS 中的模型管理也叫做分布式模型管理(Distributed

到稿日期:2012-09-04 返修日期:2012-12-01 本文受国家自然科学基金(60773188, 61142010),广东省教育部产学研合作计划引导项目(2009B090300309),中央高校基本科研业务费项目(2010MS017)资助。

黄辉(1985—),男,博士生,主要研究方向为分布式模型管理、语义网技术, E-mail: shocking_easy@163.com; 陈学广(1947—),男,教授,博士生导师,主要研究方向为信息系统工程、决策支持系统、复杂物流管理、自动协商/协调理论与方法; 王志武(1974—),男,博士,副教授,主要研究方向为决策支持系统、网格计算、云计算。

Model Management, DMM), 它主要关注的是, 怎样无缝地链接网络上的模型和数据资源^[7], 并实现传统的模型管理功能。

在模型管理的众多功能中, 模型组合无疑是最吸引人, 也是最难实现的, 尤其是在分布式环境下。从无到有地进行一项建模任务通常需要耗费大量的时间和精力, 对复杂的决策问题建模尤其如此。然而, 复杂的决策问题往往能逐层分解为不太复杂的子问题, 而这些子问题已经存在成熟的算法和模型, 因此, 一个很自然的想法是: 能否将这些已经存在的子模型通过某种机制组合起来, 形成复杂决策问题的解决方案呢? 如果可行, 就能通过共享和重用以前开发的模型, 来避免重新建模所带来的时间和金钱浪费, 从而提高模型管理的效率和质量。

随着近年来计算机网络和分布式计算技术的新发展, 以及与 DSS 相关理论和技术的结合, 模型组合的研究设想有了实现的可能, 并得到越来越多的关注。

2 模型组合概述

模型组合是问题分解的逆过程, 意味着利用模型库中已有的模型, 按照原问题的分解结构逆向构造一个模型序列, 来求解某个特殊的复杂决策问题。模型组合通常与模型选择相关联^[5], 当单个的模型满足不了问题求解的需要时, 可采用模型组合提供一套有效的解决方案。

模型组合的一个突出特点, 就是被链接起来的单个模型, 其内部结构在被选择采用后不会发生变化^[5], 这种组合是在功能层次上的模型聚集^[8]。相应地, 模型组合中用到的子模型是以“黑箱”的形式表现的, 即用户或程序无法得知和改变模型的内部结构和运行机制, 子模型的调用通过其输入输出的数据流和控制流来实现。

一般来说, 模型组合问题相当于要构造一个新的“模型”(本质上是一个控制器), 该“模型”与模型库中已有的子模型交互, 通过控制流控制子模型的调用次序, 通过数据流提交问题请求和获取子问题的解。其一般流程如图 1 所示, 图中的组合模型通过与位于 4 个模型库中的子模型交互, 来综合求得复杂问题的解, 双向箭头分别代表提交给子模型的问题请求和所求得解, 数字①到④表示调用子模型的顺序。

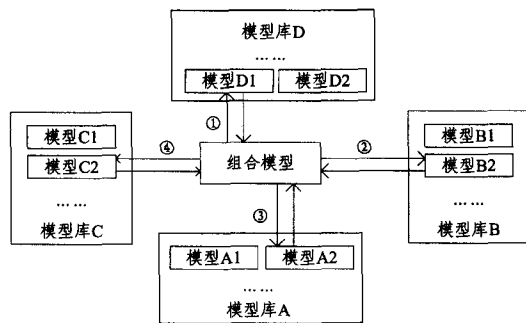


图 1 组合模型的一般形式

为了更清楚地说明组合模型的内部结构, 图 2 构造了一个组合模型的简单例子。图中的虚线箭头表示输入输出的数据流, 而实线箭头表示控制流。类似于程序设计语言, 可以利用顺序、循环和选择 3 种结构来构造任意复杂程度的控制结构, 另外, 还可加入值的判定结构等等。组合模型本身不承担或只承担少量计算任务, 主要的计算工作交给各个子模型来完成。

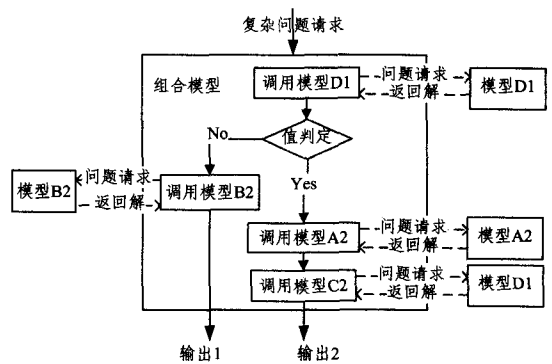


图 2 组合模型内部结构

虽说模型组合是一个很有吸引力的想法, 但是实现起来却具有相当的复杂性: ①组合模型的产生是一个从抽象到具体的过程, 需要体现出将复杂问题分而解之的逻辑性, 还要设计和编写详细的数据流程和控制流程, 如果全靠人力完成将费时费力而且容易出错; ②现实中的子模型往往是在不同时间、分别针对不同应用独立开发的, 每个子模型都可一般化地看作一个多输入多输出的系统, 因此就带来子模型间输入输出不完全匹配的问题, 即一个子模型的输出不能直接作为另一个子模型的输入, 而要经过一些中间处理; ③模型组合还涉及到一个层次性的问题, 即成功组合的多个子模型的整体可以看作一个新的子模型, 供其他的组合过程使用并形成层次结构, 由此带来的问题包括任务分解应该进行到哪一层, 这个新的子模型怎么描述它的输入输出及功能等行为; ④网络环境具有高度的动态性, 组合方案产生后, 在执行的过程中可能会因为网络阻塞, 或者存放子模型的站点故障等原因造成部分子模型不可用, 因此需要设计相应的应对策略; ⑤若需要搜索的子模型数量众多、规模庞大, 产生的组合方案所需要消耗的时间将会呈指数上升, 从而带来“状态爆炸”问题; ⑥用户提交的问题求解请求包括功能描述和非功能描述, 由于屏蔽了技术细节, 用户输入的信息往往是不完备的, 如何在信息不完备的情况下设计组合方案也是值得考虑的。

由于模型组合是模型管理的功能之一, 而且与其他功能密切联系, 例如在构造组合的过程中, 需要利用模型搜索功能来发现可用的子模型, 组合模型成功建立之后, 需要模型选择功能的配合才能完成执行, 因此离开整体功能架构来单独讨论模型组合功能是不合适的。从宏观方面来看, 为了实现模型组合, 分布式模型管理需要解决的问题主要是: ①如何发现、访问和调用不同地理位置的模型, 这也是模型的访问控制问题; ②在问题①的基础上, 如何设计模型的组合机制, 以某种逻辑或者推理的方式(而不是通过人工方式)构造组合模型(控制器), 这也是如何产生模型组合方案的问题。

3 分布式模型管理

针对分布式环境下, 不同地理位置的模型和数据资源的发现、访问和调用问题, 研究人员充分利用计算机网络技术, 提出了多种实现方式。

为支持分布式环境下的模型组合和执行, 文献^[9]扩展了文献^[10]的过滤空间(Filter Space)方法, 解决了分布在不同站点的数据资源具有重叠的关键值和不同的数据范围情况下的组合问题。作者提出了一个基于 Web 通信的模型管理框架, 为了调用不同地理位置、不同平台上的数据资源, 该框架

集成了多种数据访问方式。

文献[11]提出了一种基于 Agent 的智能模型管理(AIMM)方法, AIMM 将专家系统知识表示和推理机制结合,使得决策制定者不需要知道技术和程序的实现细节。文献[12]提出了一个利用软件代理技术建立的 DSS 框架,以便为拥有位置分布、企业全范围的、异构信息系统的组织提供支持。文章还详细介绍了构建基于 Agent 的决策支持系统的开发生命周期。

文献[13]利用了结构化建模和 XML 固有的协同作用,来促进分布式环境下的模型共享和重用,文章提出了一种基于 XML 的模式定义和两种可选的支撑框架。文献[14]提出了一个集成的四层软件框架来设计基于 Web 的分布式 DSS,并以基于组件的框架来实现上述 DSS,其中模型和数据通过组件和连接器进行管理。文献[15]回顾和总结了目前基于 Web 的 DSS 的研究和应用。

传统的基于 Web 和基于 Agent 等技术的分布式模型管理提出的时间较早,在当时的技术条件下能够解决部分问题,比如模型的远程访问和功能调用,在特殊领域、个别组织内部能建立可用的应用系统,但是也遇到了很多困难:①异构数据资源的访问需求,使得系统设计时不得不集成多种访问方法,例如 CORBA、ODBC 等等,实现非常复杂;②缺乏统一的设计规范,也没有专门的软件工具可用,由于实际问题的特殊性和复杂性,研究人员很难设计出一套通用的、被行业普遍接受的标准;③传统的模型管理方式大多是静态的管理,如果在模型组合功能执行过程中发生模型维护(删除、更新等)操作,或者某个子模型由于网络故障临时不可用,将会导致整个模型组合过程产生错误。

而由于 Web Service 技术的优越性,使得它现阶段非常适用于模型管理,以及复杂的模型组合功能的实现。Web Service 是一个松耦合的、可重用的软件组件,它语义封装各个独立的功能,是分布式的,并且能够通过标准的英特网协议编程式访问^[16]。利用 Web Service 有两个好处:首先,从技术层面来看,Web Service 代表了一系列标准协议的集合,这些协议用来创造、分配、发现和集成封装商业功能的语义软件组件;其次,从商业的角度来看,Web Service 通过集成松耦合的软件组件,提供了“刚刚好”(just-in-time)的软件服务^[5]。

Web Service 通过将模型封装为模型服务,来把模型组合问题转化为 Web Service 组合问题。Web Service 将模型看作服务,而不是数据或对象,模型通过 Web Service 标准表示和实现。决策者可以调用决策支持服务,并可在运行时组合不同数据源的组件,来得到特殊的计算支持。

文献[17]强调了 Web Service 的动态特性,设计出一套叫做 ISP&E(Integrated Service Planning and Execution)的方法,将服务组合过程与服务执行过程交错进行,来处理 Web Service 的动态和可扩展的组合问题,并提出 3 种组合失败的应对策略。文献[18]利用文献[17]中提出的 ISP&E 方法,设计了一个基于 Web Service 的模型管理框架来将模型组合和执行过程综合集成,以支持模型管理生命周期中的各种活动;并讨论了领域知识的编码问题。文献[19]分析了分布式环境下的模型管理,区分了不同类型的模型用户和各自的建模知识需求,并提出了一个基于 Web Service 的模型管理框架来管理电子表格模型。

由于一词多义和可替代词汇的存在,使得传统的基于语法(即基于关键词)的模型搜索成功率不高。为了增加模型搜索、选择的成功率,以及在模型智能组合过程中提供领域知识支持,研究人员开始探索在 Web Service 环境下,给模型的描述(用 WSDL 表示)增加语义信息。例如:文献[8]提出了一个基于语义 Web Service 的模型管理框架,作者使用了 OWL-S 来对模型进行语义标注;文献[20]也提出了一种基于语义 Web 的分布式模型管理系统的设计和实现方法,并利用 SAWSDL 和 SA-SMML 方法对不同种类的模型进行语义标注;文献[21]提出了一个语义标注模型,在用户提交请求之前,先以 Web Service 为节点、输入输出相似度值为边建立语义网络,以方便 Web Service 的发现和组合。

此外,近年来提出的网格技术(grid)是为了实现计算资源、存储资源、数据资源、信息资源、知识资源及专家资源等的全面共享与协同工作,将其应用于分布式的模型管理也是一条新颖的思路。文献[22]借鉴了 Web Service 的技术特点,将模型封装成模型服务,并通过网格的资源管理框架进行统一管理,但是未涉及模型组合功能的实现。文献[23]系统总结了目前基于网格的决策支持系统(GBDSS)的特点,提出在网格环境下进行决策服务建模要解决的 4 个方面的问题:服务组合架构、服务描述模型、组合建模机制、服务匹配和优化选择机制,并分别提出了解决方案。总的来说,将网格技术,乃至云计算用于决策支持系统毫无疑问将是未来的研究方向,但就目前来说,相关概念和技术都只是刚刚提出,还有待长期的实践检验。网格技术、云计算都是在 Web Service 技术的基础之上发展而来,Web Service 技术在 DSS 的开发和应用方面虽然取得了不少成果,但在模型组合问题方面学术界和工业界尚未达成共识。因此,继续推进完善这方面的研究,才是当务之急。

模型管理框架设计的好坏,会直接影响模型组合的效率和成败。

Web Service 环境下的模型管理框架结构大同小异,一般都需要将分布式的模型和数据库等资源封装成服务。存在争议的地方是,模型管理功能是否需要分解并封装成 Web Service 置于网络中,或是模型管理功能集中以软件客户端的形式存放于用户计算机上。前者对应“瘦客户端”,而后者对应“胖客户端”。传统的分布式模型管理系统都是采用“胖客户端”的设计方式。使用“瘦客户端”的好处是,用户只需要用浏览器就能实现模型管理的全部功能,用户计算机不需要承担额外的计算任务,但是缺点也很明显:一旦模型管理服务出现故障,对所有的终端用户而言,整个系统都将变得不可用。相反,“胖客户端”在可靠性方面要强于“瘦客户端”,但对用户计算机硬件要求更高,且需要事先在用户端部署。胖瘦客户端各有优缺点,需要按照具体系统的设计需求具体分析。

文献[18]提出了一个一般的基于 Web Service 的模型管理框架,它是对目前这方面研究的较好的归纳。

如图 3 所示, S_i 表示模型提供者, M_i 表示模型。每个提供者可提供多个模型的访问,例如 S_1 可以提供模型 M_1 、 M_2 和 M_3 的访问。同时注意到 S_3 提供预组合(pre-composed)模型的访问($M_4 \rightarrow M_8$,即由来自不同的模型提供者 S_{31} 、 S_{32} 的模型 M_4 和 M_8 组合产生的新模型)。模型的描述存放在一个集中式的注册中心。服务平台提供模型组合和执行能

力。模型组合的相关知识保存在本体、领域知识库和相关的模型容器中。

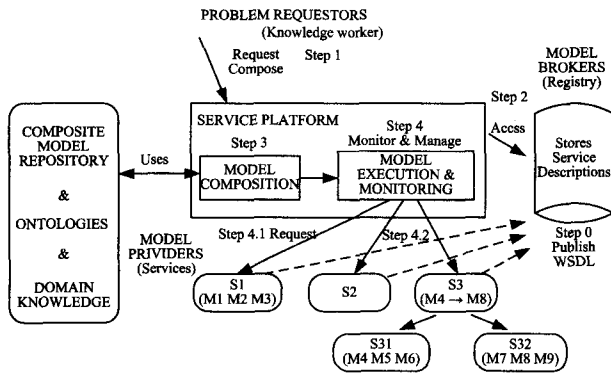


图3 基于 Web Service 的模型管理框架

问题求解的请求被发送到服务平台,基于问题的描述,平台负责从注册中心搜索合适的模型提供者,如果不存在精确的匹配,模型组合和执行功能将会被调用。

从模型管理的视角来看,此框架的主要优点包括:(1)对于一个给定的问题请求,可能存在多个可用的模型(来自不同的提供者)并且可以联合起来形成特殊任务;(2)被封装为服务的模型可以独立地维护并能以实时、标准化的方式传递;(3)该框架将软件密集型任务(编码和传递模型)与知识密集型任务(开发模型)解耦,使得不同的开发人员分工明确;(4)领域知识可被用来指导服务平台上的多种模型管理任务。

4 基于 Web Service 的模型服务组合方法比较

Web Service 本质上是无状态的、静态的,而模型组合中多个模型的交互,通常需要假设模型服务间的同步或异步信息通信是有状态的、长时间运行的,WSDL 并不能胜任传送这种类型的会话;此外,模型组合要求被选择的模型以某种结构连接和交互,即组合方案的设计,而 Web Service 规范并没有规定如何设计这种组合机制。因此,为了将单个的 Web Service 组合成具有适当复杂度的、可靠的、基于业务流程的解决方案,企业和学术界提出了多种 Web Service 组成语言和技术。由于目前分布式模型管理、组合功能的实现主要借鉴于 Web Service 组合,因此本节将兼顾讨论几类典型的 Web Service 组合技术。

实现模型组合的方法有很多,学术界对这些方法的分类也各不相同。文献[24]将目前的服务组合方法分为静态和动态两种,这种方式太过于笼统;而文献[23]将其分为基于 workflow、基于 AI 规划和基于语义匹配 3 种,其中,基于 workflow 的例子有很多,也很好区分,例如:基于 Petri 网的服务组合^[25],还有 BPEL4WS(Business Process Execution Language for Web Services),它是一个支持面向流程的服务组合的、基于 XML 的工作流语言^[26]。然而 AI 规划与语义匹配与其说是可相互替代的两大类,更像是互相配合的两种技术,毕竟 AI 规划技术主要应用于产生组合方案,而语义匹配则更多地用于模型服务选择,目前学术界经常将二者配合使用,因此,在分类学上将二者分离是不妥的。文献[27]基于输入的类型和用户参与的自由程度,将现存的解决方案分为 4 类:交错(Interleaved)组合和执行、整体(Monolithic)组合和执行、阶段组合和执行(Staged)、基于模板的(Template-based)组合和执行,

其是对大量文献的高度抽象,适用范围广泛。

本文将按智能化的程度,将目前主流的服务组合方法分为:人工实现、半自动化实现、自动化实现。所谓智能化程度,主要是指组合方案产生过程中需要人工干预的程度,可想而知,人工实现就是靠人力全程构造子模型服务间的逻辑和控制结构及其相互间通信的消息结构;而自动化的实现指的是上述过程全由计算机来处理,用户只需对最后产生的最优方案(集)作取舍;而半自动化指的是上述二者的折中,由计算机辅助用户进行设计。

人工组合的例子有很多,包括上文提到的 BPEL、基于 Petri 网之类的工作流服务组合方式,这类组合方式不但费时费力,而且容易出错。

半自动组合是一种过渡的方案。文献[28]在利用 DAML-S 和 OWL 对 Web Services 进行语义描述的基础上,提出了一个 Web Services 半自动组合框架,该框架包含一个组合器(composer)和一个推理引擎(inference engine),其中组合器在任务的每一步给用户可提供选择的服务列表,以使用户创建服务工作流。文献[29]主要描述的是基于 AI 的 Web Services 自动组合,但是作为对比,也提出了一个集成的框架,该框架同时包含人工和半自动化服务组合两种操作方式。文献[30,31]等基于模板的组合方式本质上也是半自动组合的一种,因为模板是事先已设计好的,用户选择好模板后,由模板来自动地选择服务实例,这种组合方式最适合于制定旅行计划等规则的服务组合应用。

服务组合的自动化实现是目前的研究热点,学术界提出了多种方案,其中最具有代表性的是基于 AI 规划的方式。给定行动(action)集、初始状态和要求的最终状态,AI 规划考虑如何自动产生方案。从这个角度来看,Web 服务组合问题可以看成是一个规划问题,每个服务可看作一个行动,而最终状态可看作是新的组合服务要达到的目标。AI 规划是一个大的学科分类,其中包含多种不同的规划技术,目前公认的适用于服务自动组合的规划方法有分层任务网络(Hierarchical Task Network, HTN)、模型检查(Model Checking)、事件演算(Event Calculus)、规划图(Planning-Graph)技术等。

HTN 被认为是目前最适合完成模型组合任务的规划技术之一。HTN 规划器(planner)的目的不是为了完成一个目标集合,而是完成某一任务的集合,其也叫做原始任务网络(initial task network)。规划系统的输入包括一个操作集(带有前提和后果),跟其他的规划方式类似,也包括一个方法集,每个方法都是以处方(prescription)的形式来说明怎样将某个任务分解成子任务集(更小的任务)。规划的过程就是利用方法将非原始任务重复地分解成越来越小的子任务,直到原始任务(primitive task)可使方案执行器直接执行规划操作。实质上,HTN 规划技术在状态空间进行递归搜索,并将有序任务分解和约束满足作为其搜索控制策略。比较著名的两个 HTN 规划系统是 SHOP 和 SHOP2(Simple Hierarchical Ordered Planner)。

文献[8,32]等都是利用 SHOP2 进行服务组合的例子。从这些文章中可以发现,HTN 技术往往要配合其他技术一起使用,这是由于 HTN 是在状态空间中进行遍历式的搜索,虽然详细可靠,但是如果状态空间规模庞大,搜索时间将呈指数上升,因此需要有合适的搜索策略来引导;HTN 规划所产生

的方案往往不是唯一的,因此还需要某种根据用户偏好进行方案选择的机制。HTN 的缺点就是不支持并行处理^[21],需要合理数量的领域知识^[33]等等。在文献[34]中,作者提出 HTN 是可用于模型组合的,但没有给出具体的实例。总的来说,基于 HTN 进行服务组合的特点是自顶向下,从问题的分解开始直到找到所有子问题的解。

基于模型检查的规划是一种不确定条件下的规划方法,主要目的是为了解决非决定性(non-determinism)、部分可观测(partial observability)和扩展目标(extended goals)等问题。模型检查是一种正式的验证技术,用来判定一个有限状态模型中的属性值是否成立。模型检查器和规划器通常用来实施验证。模型检查器搜寻反例来证伪一个给定的陈述,而规划器搜寻一个行动序列来满足一个目标。基于模型检查的规划从一个正式的领域模型和目标的详细描述出发,综合出一个方案^[33]。模型检查规划可以解决的问题包括:强规划(strong planning)、弱规划(weak planning)、强循环规划(strong cyclic planning)^[35]、一致性规划(conformant planning)^[36]和部分可观测条件下的强规划(strong planning under partial observability)。文献[37,38]中,作者开始尝试解决 Web Services 的自动组合问题,给定一系列原子服务的抽象描述(BPEL 描述)和一个组合请求,利用模型检查规划技术可以自动产生一个可执行的 BPEL 过程。文献[39]提出了一个新颖的 Web Service 组合框架,并正式给出了该组合框架的基本原理,通过将 Web Service 表述为有限状态机,再引入模型检查规划方法作为推理机制,来自动产生组合方案。该方法需要事先给出组合模型的抽象描述,然后通过规划的方式产生所有的交互细节,从而得到一个可执行的组合服务,因此可被用于已知组合模型的抽象结构的情况,而不适用于完全自动产生组合方案。

事件演算最初由 Kowalski 和 Sergot^[40]提出,是相对于“情景演算”的一个概念。二者的主要区别是概念上的:情景演算处理全局状态,而事件演算处理局部事件和时间段。事件演算是一种时间的形式化(temporal formalism),用来对情节进行建模和推理,情节以事件集的形式描述。它的主要特色是在经典的一阶逻辑中引入了时间的概念。文献[41-43]发展和完善了最初的事件演算概念,将其简化并引入溯因(abductive)机制,使其适用于服务组合。文献[44,45]即是事件演算在不同组合方式下的应用。

此外,还有一些特殊的方法也被提出用于解决服务自动组合问题,例如背景感知(context-aware)^[46,47]、反向链^[21]、过滤空间(Filter Space)^[9,10]等等。

学术界提出了多种服务自动组合的技术和方法,它们虽然能应对各种特殊的应用场合,但都只能部分解决第二节中罗列的问题,目前各种方法的缺陷也很明显:HTN 和事件演算方法不适应具有高度动态性的网络环境,模型检查只适用于已知组合模型的抽象结构的情况;背景感知虽然在应对网络动态性方面有优势,但只是针对单个组合方案;反向链需要对已存在的服务预处理才能使用,只适用于小规模网络;过滤空间产生的组合方案是固态的,无法局部修改。此外,上述所有自动组合方法的共同点是需要在问题领域和规划领域之间作两次映射(组合方案在规划领域中产生,在问题领域中执行),且都无法很好地应对大规模的服务环境(都会产生“状态爆炸”问题)。

更重要的是,从上述的综述中可以看出,大部分的自动组合技术都是针对 Web Service 提出的,只有很少的研究是专门针对分布式模型组合^[34]。模型组合有其自身的特点,传统的 Web Service 自动组合方法只适用于定量模型组合的情况。一般来讲,DSS 模型包括定量和定性模型,定量模型主要是以量化的、数字的形式来表述,其输入输出都是数字形式;而定性模型是大部分以文字描述为主,辅以推理,其输入输出多为定性的字符描述(可能伴随有部分定量内容)。换句话说,如果需要求解的复杂问题同时涉及到定量和定性模型,那么传统的 Web Service 自动组合方法将无法处理,原因是二者的输入输出不能直接相连。因此,设计专门针对分布式 DSS 环境下的模型组合机制,还需学术界的进一步努力。

结束语 模型组合是 DSS 模型管理的重要功能之一,其相关概念和设想很早就已经提出,但是限于当时的技术条件一直未能很好地实现。随着近年来网络和人工智能规划等技术的发展,以往难以解决的模型自动组合问题有了实现的可能。另外,工业界和学术界在这一领域的研究并没有达成共识,相关标准也还不完善,因此,模型组合仍将是 DSS 模型管理研究的热点和难点。

近年来,基于语义 Web Service 的自动组合研究得到了众多的关注,取得了不少成果,但主要的应用还局限于电子商务、旅行预订等简单应用,真正将其与 DSS 模型管理相联系的研究和应用还不是很多。正如第 2 节所讲,模型组合涉及到一系列要解决的问题,其中最重要的,就是要体现模型组合的智能性、通用性和动态适应性。目前几乎所有的 Web Service 自动组合方法都只针对某个特殊领域,而不适合在大规模、跨领域、高动态性的网络环境中应用。更重要的是,传统的 Web Service 组合方法并不完全适用于 DSS 模型组合的情况。如何将传统的方法扩展和完善,或者提出新的思路,以促进 Web Service 环境下模型自动组合的研究和应用,将是未来有待深入研究的问题。

参考文献

- [1] Sprague R H, Carlson E D. Building effective decision support systems[M]. Prentice-Hall, 1982
- [2] Elam J J, Henderson J C, Miller L W. Model Management Systems: An Approach to Decision Support in Complex Organizations[C]// Proceedings of the First Conference on Information Systems. 1980
- [3] Bonczek R H, Holsapple C W, Whinston A B. A Generalized Decision Support System Using Predicate Calculus and Network Data Base Management[J]. Purdue University, 1981, 29(2): 263-281
- [4] Blanning R. A Relational Framework for Model Management In Decision support Systems[Z]. DSS-82 Transactions, 1982
- [5] Liang T-P, Lee C-C, Turban E. Model Management and Solvers for Decision Support[J]. Handbook on Decision Support Systems, 2008, 1: 231-258
- [6] Ching C, Holsapple, Whinston A. Computer Support in Distributed Decision Environments[Z]. Environments for Supporting Decision Processes, 1991
- [7] Huh S-Y, Kim H-M, Chung Q B. Framework for change notification and view synchronization in distributed model management systems[J]. Omega, 1999, 27(4): 431-443
- [8] Deokar A V, El-Gayar O F. A Semantic Web Services-Based Ar-

- chitecture for Model Management Systems[C]//Hawaii International Conference on System Sciences, Proceedings of the 41st Annual. 2008
- [9] Chari K. Model composition in a distributed environment[J]. *Decision Support Systems*, 2003, 35(3): 399-413
- [10] Chari K. Model Composition Using Filter Spaces[J]. *Information Systems Research*, 2002, 13(1): 15-35
- [11] Liu J I, Yun D Y Y, Klein G. An Agent for Intelligent Model Management[J]. *Journal of Management Information Systems*, 1990, 7(1): 101-122
- [12] Bui T, Lee J. An agent-based framework for building decision support systems[J]. *Decision Support Systems*, 1999, 25(3): 225-237
- [13] El-Gayar O, Tandekar K. An XML-based schema definition for model sharing and reuse in a distributed environment[J]. *Decision Support Systems*, 2007, 43(3): 791-808
- [14] Zhang S, Goddard S. A software architecture and framework for Web-based distributed Decision Support Systems[J]. *Decision Support Systems*, 2007, 43(4): 1133-1150
- [15] Bhargava H K, Power D J, Sun D. Progress in Web-based decision support technologies[J]. *Decision Support Systems*, 2007, 43(4): 1083-1095
- [16] Gottschalk K, et al. Introduction to Web Services Architecture [J]. *IBM Sys J*, 2002, 41(2): 170-177
- [17] Madhusudan T, Uttamsingh N. A declarative approach to composing web services in dynamic environments[J]. *Decision Support Systems*, 2006, 41(2): 325-357
- [18] Madhusudan T. A Web services framework for distributed model management[J]. *Information Systems Frontiers*, 2006, 9(1): 9-27
- [19] Iyer B, Shankaranarayanan G, Lenard M L. Model management decision environment; a Web service prototype for spreadsheet models[J]. *Decision Support Systems*, 2005, 40(2): 283-304
- [20] Deokar A V, El-Gayar O F, Aljafari R. Developing a Semantic Web-Based Distributed Model Management System: Experiences and Lessons Learned[C]//System Sciences (HICSS). 2010 43rd Hawaii International Conference on. 2010
- [21] Talantikite H N, Aissani D, Boudjlida N. Semantic annotations for web services discovery and composition [J]. *Computer Standards & Interfaces*, 2009, 31(6): 1108-1117
- [22] 王志武. 基于网格的DSS模型服务管理研究 [D]. 武汉: 华中科技大学, 2007
- [23] 王煜. 基于MDA的GBDSS决策服务组合建模方法研究 [D]. 武汉: 华中科技大学, 2010
- [24] 李景霞, 侯紫峰. Web服务组合综述[J]. *计算机应用研究*, 2005(12): 4-7
- [25] Hamadi R, Benatallah B. A Petri Net-based Model for Web Service Composition[C]//Proceedings of the 14th Australasian Database Conference. 2003
- [26] OASIS. Web services business process execution language version 2. 0[OL]. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>, 2007
- [27] Agarwal V, et al. Understanding approaches for web service composition and execution[C]//Proceedings of the 1st Bangalore Annual Compute Conference. 2008
- [28] Sirin E, Hendler J, Parsia B. Semi-automatic Composition of Web Services using Semantic Descriptions[C]//Web Services: Modeling, Architecture and Infrastructure Workshop in ICEIS. 2003
- [29] Digiampietri L A, Perez-Alcazar J J, Medeiros C B. AI Planning in Web Services Composition; a review of current approaches and a new solution[OL]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.2019,2012>
- [30] Lee S, et al. OntoPipeliner: An ontology-based automatic semantic service pipeline generator[J]. *Expert Systems with Applications*, 2011, 38(8): 9472-9482
- [31] Lécué F, et al. SOA4All: An Innovative Integrated Approach to Services Composition[C]//2010 IEEE International Conference on Web Services. 2010
- [32] Sirin E, et al. HTN planning for Web Service composition using SHOP2[J]. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2004, 1(4): 377-396
- [33] Chan K S M, Bishop J, Baresi L. Survey and Comparison of Planning Techniques for Web Services Composition[M]. University of Pretoria. © ISMED, 209, 2007
- [34] Deokar A V, El-Gayar O F. Enabling Distributed Model Management Using Semantic Web Technologies[C]//System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on. 2009
- [35] Cimatti A, et al. Weak, strong, and strong cyclic planning via symbolic model checking[J]. *Artificial Intelligence*, 2003, 147(1/2): 35-84
- [36] Cimatti A, Roveri M. Conformant planning via symbolic model checking[J]. *J. Artif. Int. Res.*, 2000, 13(1): 305-338
- [37] Pistore M, et al. Automated composition of web services by planning at the knowledge level[C]//Proceedings of the 19th International Joint Conference on Artificial Intelligence 2005. Edinburgh, Scotland; Morgan Kaufmann Publishers Inc., 2005: 1252-1259
- [38] Pistore M, et al. Automated Synthesis of Composite BPEL4WS Web Services[C]//Proceedings of the IEEE International Conference on Web Services 2005. IEEE Computer Society. 2005: 293-301
- [39] Bertoli P, Pistore M, Traverso P. Automated composition of Web services via planning in asynchronous domains[J]. *Artificial Intelligence*, 2010, 174(3/4): 316-361
- [40] Kowalski R A, Sergot M J. A logic-based calculus of events[J]. *New Gener. Comput.*, 1986, 4(1): 67-95
- [41] Shanahan M. Prediction is deduction but explanation is abduction[C]//Proceedings IJCAI 89. 1989; 1055-1060
- [42] Shanahan M. Event calculus planning revisited[C]//Proceedings 4th European Conference on Planning (ECP 97). Springer Lecture Notes in Artificial Intelligence no. 1348, 1997; 390-402
- [43] Shanahan M. An abductive event calculus planner[J]. *The Journal of Logic Programming*, 2000, 44(1-3): 207-240
- [44] Okutan C, Cicekli N K. A monolithic approach to automated composition of semantic web services with the Event Calculus [J]. *Knowledge-Based Systems*, 2010, 23(5): 440-454
- [45] Ozorhan K, E, Kuban E K, et al. Automated composition of web services with the abductive event calculus[J]. *Information Sciences*, 2010, 180(19): 3589-3613
- [46] Li L, Liu D, Bouguettaya A. Semantic based aspect-oriented programming for context-aware Web service composition[J]. *Information Systems*, 2011, 36(3): 551-564
- [47] Niu W, et al. Multi-granularity context model for dynamic Web service composition[J]. *Journal of Network and Computer Applications*, 2011, 34(1): 312-326