多主体系统开发环境的结构化评价框架SEF及评价结果*)

陈嘉佳　毛新军

(国防科学技术大学计算机系　长沙410073)

摘　要　近年来学术界和工业界推出了大量的多主体系统开发环境(MASDE: Multi-Agent System Development Environment)。如何评价已有的各种MASDE,帮助软件开发人员从大量、异构和多样化的MASDE中选择"合适"的MASDE以及促进MASDE的标准化和集成化是目前人们面临的一项重要研究课题。本文从工程实践的角度提出了一个结构化的MASDE评价框架SEF;并对一组典型的MASDE作了深入的分析和评价,得到了一系列重要的评价结果。与已有工作相比较,SEF具有多视角、易于操作、可扩展以及独立于应用等特点,能较好地用于描述、分析和展示MASDE的基本特征、发现其优势和不足。

关键词　主体,多主体系统,多主体系统开发环境,结构化评价框架

A Structural Evaluation Framework and Results of Multi-Agent System Development Environment

CHEN Jia-Jia　MAO Xin-Jun

(Dept. of Computer Science, National University of Defense Technology, Changsha 410073)

Abstract　Recently many Multi-Agent System Development Environments (MASDEs) have been developed. Therefore, in the literature of AOSE, it has become an open issue to evaluate the various MASDEs in order to help the developers select the proper MASDE from different MASDEs, and promote the standard and integration of MASDEs. In this paper, a structural evaluation framework (SEF) is presented, which have such characteristics as multi-perspective, easy-to-use, extensible and independence of application. Based on SEF, a number of typical MASDEs are selected and evaluated, and some important evaluation results are obtained.

Keywords　Agent, Multi-Agent system, MASDE, Structured evaluation framework

1　引言

近年来,随着主体研究的不断发展,学术界和工业界都迫切希望能够将面向主体的软件开发技术应用于实际系统的开发,尤其是希望作为一种主流的软件开发技术推动复杂软件系统的工业化开发。同时,研究人员已经认识到,缺乏功能强大、灵活多样的开发工具和环境的支持是制约主体技术应用的重要因素。因此,作为促进主体技术广泛应用的主要推动力之一,多主体系统开发环境的研究和发展受到了业界的高度重视。本文所指的多主体系统开发环境(MASDE: Multi-Agent System Development Environment)是一个外延较为广泛的概念,包括主体开发包和可重用库、主体框架、主体原型开发环境、主体开发工具集以及主体基础设施和主体平台等。

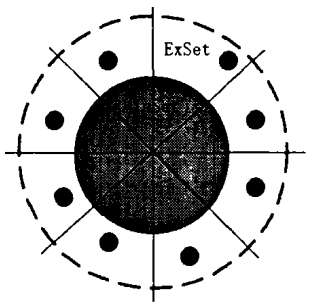
经过20多年的发展,迄今为止人们已经推出了数十种MASDE,并且各种新的MASDE还在不断涌现。尽管如此,由于目前主体技术仍然缺少可供广泛接受的定义和标准,不同的MASDE往往具有不同的技术背景,表现出不同的技术优势和特点,具有不同的应用领域,以至所开发的系统通常是异构的(使用不同的框架、工具和语言实现,运行在不同的平台上),难以实现集成和互操作。因此,如何评价已有的各种MASDE,帮助主体系统开发人员从大量、异构和多样化的MASDE中选择"合适"的MASDE以及促进MASDE的标准化和集成化就成了目前学术界和工业界面临的一个重要研究课题。针对这个问题,本文提出了一个结构化的MASDE评估框架SEF来对这些MASDE进行系统、深入的比较、分析和评价。

本文第2节介绍了SEF的结构和内容,第3节基于SEF对一组典型的MASDE进行了深入的剖析,列举了具体的评价结果,最后介绍了相关工作,给出了相应的结论。

2　多主体系统开发环境的结构化评价框架SEF

2.1　SEF的结构

SEF是一个结构化的MASDE评价框架,由三个不同抽象层次的元素组成:组、核心概念和评价条目(如图1所示)。



每个扇形代表一个组,内圆表示最小集MinSet,外圆和内圆之间的环形表示扩展集ExSet。外圆的虚线表示可扩展,内圆的实线表示封闭。每个黑色实心圆代表一个组的核心概念。

图1　SEF的结构

·组　一个组代表了MASDE分析和评价的一个视角,用于衡量MASDE在某方面所达到的高度。SEF由多个不同的组构成,也就是从多个不同的角度来分析MASDE,从而可以得到更加完整而全面的结果。

·核心概念　一个组具有一个核心概念,用以说明MAS-

*)本课题得到国家自然科学基金项目(批准号60373022)资助。陈嘉佳　硕士生,研究方向为面向主体的软件工程。

DE 在该视角应达到的高层目标。比如，体系结构组的核心概念是“通用性”，也就是说 MASDE 在体系结构上应使得所开发的主体系统满足“通用性”的要求。因此，核心概念是对一个 MASDE 评价视角的高层抽象，也是 MASDE 在该方面的理想目标，但需使用一组具体的评价条目来描述如何达到这个抽象目标。

·评价条目 一个小组由一个核心概念和一组与该概念有关的评价条目组成。核心概念描述了 MASDE 在该组应该达到的理想目标，而评价条目描述了在该组对 MASDE 进行比较和分析的实际尺度。核心概念是大粒度、高层、抽象的目标，而评价条目是小粒度、底层、具体的衡量尺度，且更加易于描述、判断和分析。比如，在体系结构组，与核心概念“通用性”相对应的评价条目集合为{反应式，意图式，社会式，移动式，多主体系统体系结构}，也就是 SEF 将根据这五个条目来具体分析某个 MASDE 所开发的主体系统在体系结构上的特点，以及是否满足“通用性”的要求。

为了更好地将评价标准集结构化，SEF 还将其评价条目分为两部分：最小集 MinSet 和扩充集 ExSet。MinSet 中的元素是对 MASDE 基本、关键属性的评价，而 ExSet 中的元素是对 MASDE 高级属性的评价。在 SEF 中，每个组的 MinSet 是相对封闭和稳定不变的，而 ExSet 则可以根据应用领域的不同进行必要的改变和扩展。

设 GROUP 是所有组的集合，CONCEPT 是所有核心概念的集合，ITEM 是所有评价条目的集合，$\mathscr{P}$ 是幂集符号。因此，SEF＝{GROUP，CONCEPT，ITEM，Concepts，Items}，其中 Concepts：GROUP→CONCEPT，Items ＝ MinSet ∪ ExSet，MinSet：GROUP × CONCEPT→$\mathscr{P}$(ITEM)，ExSet：GROUP×CONCEPT →$\mathscr{P}$(ITEM)。

2.2 SEF 的组成

SEF 共分8个组，分别从8个不同的视角来对 MASDE 进行详细的分析和评价：体系结构、通信与交互、兼容性、面向主体的软件工程、工具集、文档、主体平台以及开发商，即 GROUP ＝{Architecture，Commuincation&Interaction，Compatibility，AOSE，Platform，ToolSet，Documentation，Producer}。其中前5组是 MASDE 特有的衡量标准，而后3组是所有软件开发环境通用的衡量标准。

1. 体系结构组 该组的核心概念是“通用性”，即 Concepts(Architecture)＝{通用性}。主体的体系结构定义了主体的内部组成以及行为决策形成过程。不同的体系结构定义了不同的组成和决策过程，强调主体系统的不同特征。而通用的体系结构兼有各种基本体系结构的特点，适用于不同的应用，因而是关键的衡量标准。

该组的最小集为 MinSet(Architecture，通用性)＝{反应式，意图式，社会式，移动式}。主体系统的体系结构可以分为两个层次：单主体体系结构和多主体体系结构。反应式、意图式、社会式以及移动式是四种最基本的单主体体系机构。

该组的扩充集为 ExSet(Architecture，通用性)＝{多主体体系结构}。多主体体系结构是指多个主体如何构成一个软件系统。

2. 通信与交互组 该组的核心概念是“社会性”，即 Concepts(Commuincation&Interaction) ＝{社会性}。社会性是主体的最重要和基本的特征之一。主体之间的通信与交互大体上可分为三个层次：首先是消息传输机制，用于底层消息的传递；第二是主体通信语言(ACL)，定义了交互消息的格式(语法)和语义以及支持参与交互的主体对这些消息的理解和分析；第三是主体交互协议，规定了交互的对象和行为以及交互的时序，从而实现复杂的交互和合作。

该组的最小集 MinSet(Commuincation&Interaction，社会性)＝{消息传输机制，主体通信语言，主体交互协议}。

该组的扩充集 ExSet(Commuincation&Interaction，社会性) ＝{本体论，内容描述语言}。本体论用于定义应用域中的各种概念、术语和关系。主体通信语言本身只定义了交互消息的格式，而其中的内容描述语言才定义了消息的具体内容。

3. 兼容性组 该组的核心概念是“互操作性”，即 Concepts(Compatibility) ＝{互操作性}。当前由于 MASDE 种类繁多，使用不同 MASDE 所构造的多主体系统往往是异构的，包括实现异构、框架异构、通信异构和平台异构等方面，导致难以实现多主体系统之间的集成和互操作。因此，互操作性是衡量 MASDE 的一项重要标准。

该组的最小集 MinSet(Compatibility，互操作性)＝{FIPA00001}。互操作性主要通过各种主体技术标准来保证，而 FIPA00001[5](FIPA 抽象体系结构规范)是目前被广泛接受的基本的主体互操作标准，它在抽象层次上定义了两个主体是如何通过注册主体和交换消息来定位和相互通信的。

该组的扩充集 MinSet(Compatibility，互操作性) ＝{FIPA00023，MASIF}。FIPA00023[6](FIPA 主体管理规范)提供了一个主体管理参考模型，包括主体创建、注册、定位、通信、迁移和撤退的逻辑框架。MASIF 规范[7]主要规定了如何实现移动主体之间的互操作性。

4. 面向主体的软件工程组 该组的核心概念是“良定义的软件工程”，即 Concepts(AOSE)＝{良定义的软件工程}。MAS 的开发是一个相当复杂的问题，因此采用一个良定义的软件工程来指导软件开发是十分重要的。

该组的最小集 MinSet(AOSE，良定义的软件工程)＝{分析，设计，开发，部署}。对于良定义的软件工程而言，面向主体开发方法定义了以主体为基本概念和主要构成元素来进行软件开发的方法学，是非常重要的组成元素。而 MASDE 所使用的开发方法是否覆盖分析、设计、开发和部署等软件开发的基本步骤也就成了 MASDE 的重要衡量标准。

该组的扩充集 ExSet(AOSE，良定义的软件工程)＝{主体实现语言，主体建模语言}。在实际开发中，使用什么编程语言来实现多主体系统非常重要。MASDE 使用统一的主体建模语言可以更加有效地建模和描述多主体系统的结构和行为。

5. 主体平台组 该组的核心概念是“可重用性”，即 Concepts(Platform)＝{可重用性}。主体平台的良好可重用性可以提高多主体系统开发的效率和质量。

该组的最小集 MinSet(Platform，可重用性)＝{分布，通信服务，可重用库}。在当前网络环境下，分布能力是对主体平台的基本要求之一。统一的通信服务可以用于实现主体之间的通信，乃至不同主体平台之间的通信，因此也是对主体平台的基本要求之一。一个功能强大的、可重用程度高的组件库可以在很大程度上减轻多主体系统的开发努力和提高开发效率。

该组的扩充集 ExSet(Platform，可重用性) ＝{主体管理服务，目录服务，安全支持，持久化支持，主体监控机制，系统日志，遗留系统集成能力}。主体管理服务一般由主体管理系统提供，而主体管理系统是主体平台的必要组成部分，用于管理主体的创建、删除、迁移等操作。目录服务一般分为两种：白页服务提供一种定位单个主体的方法，即注册服务，而黄页服务则提供了一种通过给定类别来定位主体的方法，即广播服务。主体平台应该能够提供统一的安全服务。持久化支持是

指把主体的信息保存在数据库中，在系统崩溃之后或主体出错之后恢复主体运行。主体的许多运行情况在设计阶段是不可确定的，因此主体平台应能提供监控机制，用于主体的调试和跟踪。许多实际应用都需要记录系统的运行轨迹和日志，因此需要对主体平台的这种能力进行评估。由于目前已有大量的遗留系统，因此主体平台对遗留系统的集成能力也是非常重要的。

6. 工具集组　该组的核心概念是"有效性和易用性"，即Concepts(ToolSet)={有效性和易用性}。易用性是针对初级用户而言的，应该允许他们相对容易地创建和部署多主体系统。有效性是针对高级用户而言的，应该允许他们有效地访问内部组件和创建复杂的多主体系统。

该组的最小集MinSet(ToolSet，有效性和易用性)={可视化的工具集，从设计到实现的转变，工具集对主体开发方法的支持}。工具集的可视化可以提高其易用性，因而常常成为衡量工具集的一个重要指标。从设计到实现的转变这个角度来看，各个不同的MASDE所采用的工具和方法各不相同，因此对其评价相当困难。本文将其分为两类：编程型(即通过"编码－编译－运行"的方式来将设计转变为实现)和设计型(即通过"填充和选择GUI－自动生成代码－编译－运行"的方式来将设计转变为实现)，然后考察每类MASDE提供的工具集在这个转变中所起的作用。此外，该组还考察MASDE是否在工具层面上为主体开发方法提供良好的支持。

该组的扩充集ExSet(ToolSet，有效性和易用性)＝{安装与配置，工具集完备性，知识基础}。安装与配置的简易与否直接影响系统的易用性。工具集完备性主要考察MASDE的工具集对于开发多主体系统来说是否完备，比如说有没有主体调试工具等。知识基础是指初级用户使用这个MASDE的工具集需要补充多少知识，用于衡量工具的易用性。

7. 文档组　该组的核心概念是"易理解性"，即Concepts(Documentation)={易理解性}。一组完备而易于理解的文档对于开发人员掌握和使用该MASDE是极为重要的。

该组的最小集MinSet(Documentation，易理解性)={用户使用指南，用户开发指南，可重用库文档}。用户使用指南用于指导初级用户如何使用该MASDE。用户开发指南用于指导高级用户如何使用MASDE开发多主体系统。可重用库文档用于描述可重用库的组成部分、各部分的关系以及如何使用各个部分来开发面向主体系统。

该组的扩充集ExSet(Documentation，易理解性)＝{实例}。丰富而逐步深入的实例以及对实例的详尽解释将十分有助于用户快速掌握MASDE。

8. 开发商组　该组的核心概念是"良好的技术支持"，即Concepts(Producer)={良好的技术支持}。一个好的开发商应该为其产品提供良好的技术支持。

该组的最小集MinSet(Producer，良好的技术支持)={不断有新版本和补丁推出，邮件列表}。对于一个好的产品而言，应该不断有新的版本推出，一旦发现bug，也要及时推出补丁。邮件列表用于评价面向主体软件开发环境的开发组织是否维护一个经常性更新的邮件列表。

该组的扩充集ExSet(Producer，良好的技术支持)={咨询和培训，第三方技术支持}。开发商愿意提供一定的咨询和培训总是一件好事。第三方技术支持是指当MASDE没有提供开发者需要的功能时，是必须自己开发一个还是可以需求其它技术支持。

总之，整个SEF的最小集共23个条目，扩充集共20个条目，共计43条目。

3　对典型多主体系统开发环境的评价结果

本节基于SEF对所挑选的10个MASDE进行系统、深入的分析和评价。所遵循的挑选准则为：1)一直在发展之中，不断有新版本推出；2)每个MASDE都有其明显的技术特点，比如AgentBuilder和Zeus以良好的工具集著称；3)兼顾有较长历史的面向主体软件开发环境和最新出现的面向主体软件开发环境。最终所选择的是：ADK、AgentBuilder、AgentFactory、AgentTool、DECAF、Grasshopper、JACK、JADE 、MadKit和Zeus(见表1)。各个MASDE更详细的内容请参看文中标出的相关参考文献。

表1　典型MASDE简介

MASDE	开发者	类型	评估版本	简介
ADK[8]	Tryllian公司	商业	ADK 2.1	专门开发移动主体的商业平台
AgentBuilder[9]	IntelliOne Technologies公司	商业	AgentBuilder 1.3 Pro Evaluation	主体开发工具套件
AgentFactory[10]	爱尔兰Dublin大学计算机PRISM实验室	学术	AgentFactory Evaluation 0.8	多主体系统开发环境
AgentTool[11]	美国Kansas State大学计算与信息科学系MultiAgent & Cooperative Robotics Lab	学术	AgentTool 1.8.3	支持使用面向主体开发方法MaSE开发多主体系统的可视化工具
DECAF[12]	美国Delaware大学信息与计算机科学系J. Graham和K. Decker	学术	DECAF 2.2.1	主体操作系统，提供了构造大粒度智能主体所需的各种服务
Grasshopper[13]	IKV++Technolgies AG公司	商业	Grassphopper Core System 2.2	基于Java的移动主体平台，支持OMG MASIF标准和FIPA 97标准
JACK[14]	澳大利亚Agent Oriented Software公司	商业	JACK 4.1	一个使用组件的方法构建、运行、集成商业级多主体系统的开发环境
JADE[15]	Italia TELECOM LAB实验室	商业	JADE 3.0b1	使用Java语言实现的软件框架，通过遵循FIPA规范的中间件和工具集来简化多主体系统的实现
MadKit[16]	MadKit兴趣小组	学术	MadKit 3.1b5	一个基于AGR组织模型的通用主体平台
Zeus[17]	British Telecommunication plc.公司	商业	Zeus 1.2.1	致力于多主体系统快速设计、开发和部署的软件组件库和工具

3.1 分类评价

给定评价框架之后，如何进行评价有多种方法，比如定性或定量的方法。由于MASDE评价的复杂性，本文采取一种较简单的定量方法，即每个条目满分为1分，只有三个等级(0分，表示不满足该条目要求，0.5分，部分满足该条目要求，1分，较好地满足该条目要求)，总计43分。某些条目评价标准略有不同，下文中已有具体说明。

1. 体系结构视角　该组的评价结果如表2所示，满分5分。从体系结构的视角来看，大多数MASDE所支持的单主体体系结构都是若干种基本体系结构的混合体，兼有它们的特点。尽管如此，到目前为止仍然不存在兼有四种基本体系结构特点的通用体系结构，值得MASDE开发者注意。另外，大多数MASDE还缺乏对多主体体系结构的有效支持。

2. 通信和交互视角　该组的评价结果如表3所示，满分5分。从通信和交互的视角来看，AgentTool、DECAF、JACK和MadKit属于一般层次(0～1分)，而ADK、Grasshopper和JADE属于中等层次(2～3分)，AgentBuilder、AgentFactory以及Zeus属于较好层次(4～5分)。MASDE所使用的消息传输机制一般都基于常见的TCP/IP、RMI或CORBA等协议，所使用的主体通信语言主要是KQML和FIPA ACL，表明在这两个方面已有共识，但在其他方面(如主体交互协议、本体论以及内容描述语言)还需进一步标准化。

表2　体系结构视角评价结果

体系结构组	最小集				扩充集	小结
	反应式	意图式	社会式	移动式	多主体体系结构	
ADK	1	0	0	1	0	2
AgentBuilder	0	1	1	0	0	2
AgentFactory	0	1	1	1	0	3
AgentTool	0	0	0	0	0	0
DECAF	1	0	0	0	0	1
Grasshopper	0	0	0	1	0	1
JACK	1	1	0	0	1	3
JADE	1	0	0	1	0	2
MadKit	0	0	0	0	1	1
Zeus	0	1	1	0	0	2

表3　通信与交互视角评价结果

通信与交互	最小集			扩充集		小结
	消息传输机制	主体通信语言	主体交互协议	本体论	内容描述语言	
ADK	1(JXTA)	1	0	0	0	2
AgentBuilder	1(RMI、CORBA、TCP/IP)	1(KQML)	1	1	0	4
AgentFactory	1(多种网络协议)	1(FIPA ACL)	1	1	0	4
AgentTool	0	0	0	0	0	0
DECAF	0	1(KQML)	0	0	0	1
Grasshopper	1(CORBA IIOP、Java RMI、TCP/IP套接字连接)	1(使用插件支持FIPA ACL)	0	0	0	2
JACK	1(UDP、DCI、CORBA)	0	0	0	0	1
JADE	1(RMI、CORBA IIOP)	1(FIPA ACL)	0	1	0	3
MadKit	1(分布主体之间通信由特定通信主体实现)	0	0	0	0	1
Zeus	1(TCP/IP套接字连接)	1(KQML)	1	1	1	5

3. 兼容性视角　该组的评价结果如表4所示，满分3分。Grasshopper是第一个遵循OMG MASIF和FIPA 97规范的MASDE。JADE与FIPA 2000规范兼容。其他MASDE则不支持互操作性要求。

表4　兼容性视角评价结果

兼容性组	最小集	扩充集		小结
	FIPA00001	FIPA00023	MASIF	
ADK	0	0	0	0
AgentBuilder	0	0	0	0
AgentFactory	0	0	0	0
AgentTool	0	0	0	0
DECAF	0	0	0	0
Grasshopper	1	1	1	3
JACK	0	0	0	0
JADE	1	1	0	2
MadKit	0	0	0	0
Zeus	0	0	0	0

4. 面向主体的软件工程　该组的评价结果如表5所示，满分6分。需要注意的是，对“主体实现语言”而言，如果MASDE使用面向主体的程序设计语言，得1分，否则得0分；对“主体建模语言”而言，目前还不存在通用的主体建模语言，因此如有主体建模语言，得0.5分，否则0分。

从面向主体的软件工程的角度来看，现有的MASDE都有所欠缺，大多数MASDE都缺乏良定义的开发方法的支持，更缺乏可视化工具对开发方法的支持。通过分析可以看出，Java尽管并不是理想的主体实现语言，但已经成为主流的主体实现语言。就目前而言，主体建模语言的研究还处于起步阶段。

5. 主体平台视角　该组的评价结果如表6所示，满分10分。需要注意的是，对“遗留系统集成能力”而言，如果只能使用Java语言来集成遗留系统，得0.5分。

从主体平台的角度来看，AgentBuilder和AgentTool所提供的主体平台的可重用性较差(0～3分)；DECAF、JACK、JADE、Madkit以及Zeus所提供的主体平台处于中等程度(4～7分)；而ADK、AgentFactory和Grasshopper则提供了较好的主体平台(8～10分)。可以看到，商业产品尤其是商业移动主体平台(如ADK和Grasshopper)比较注意提供较为完善的服务，而平台可以显著提高主体的可重用性、健壮性和可靠性。

表5 面向主体的软件工程视角评价结果

体系结构组	最小集				扩充集		小结
	分析	设计	开发	部署	主体实现语言	主体建模语言	
ADK	0	0	0	0	0(Java)	0	0
AgentBuilder	1	1	1	1	1(RADL)	0	5
AgentFactory	1	1	1	1	1(Agent Factory Agent Programming Language)	0.5	5.5
AgentTool	1	1	0	0	0	0.5	2.5
DECAF	0	0	0	0	0(Java)	0	0
Grasshopper	0	0	0	0	0(Java)	0	0
JACK	0	0	0	0	1(JACK Agent Language)	0.5	1.5
JADE	0	0	0	0	0(Java)	0	0
MadKit	0	0	0	0	0(Java、多种脚本语言)	0	0
Zeus	1	1	1	1	0(Java)	0	4

表6 主体平台视角评价结果

通信与交互	最小集			扩充集							小结
	分布	通信服务	可重用库	主体管理服务	目录服务	安全支持	持久化支持	主体监控机制	系统日志	遗留系统集成能力	
ADK	1	1	1	1	1	1	1	0.5	1	0.5	9
AgentBuilder	1	1	0	0	0	0	0	0	0	0	2
AgentFactory	1	1	1	1	1	1	1	1	1	0.5	9.5
AgentTool	0	0	0	0	0	0	0	0	0	0	0
DECAF	1	1	1	0	1	0	0	0	0	0.5	4.5
Grasshopper	1	1	1	1	1	1	1	1	0	0.5	8.5
JACK	1	1	1	0	0	0	0	0.5	0	0.5	4
JADE	1	1	1	1	1	0	0	1	0	0.5	6.5
MadKit	1	1	1	1	1	0	0	1	0	0.5	6.5
Zeus	1	1	1	0.5	1	0	0.5	1	0	0.5	6.5

6. 工具集视点 该组的评价结果如表7所示，满分7分。工具集的有效性和易用性有时是一对矛盾，这是因为有效性强的工具集往往需要使用者对工具集的技术背景有深入的了解，对工具集本身有透彻的理解，对于一般用户来说其用法过于复杂。对于 AgentBuilder、AgentFactory、AgentTool、DECAF、Zeus 等 MASDE 所提供的可视化工具集来说，虽然其工具集很有效(比如能够自动生成主体代码)，但较为难用；而对于 Grasshopher、JACK、JADE 等 MASDE 所提供的工具集来说，虽然用处不大(这些开发环境主要通过编程来实现主体)，但挺好用；而 ADK、MadKit 则有效性和易用性均不足。

表7 工具集视角的评价结果

通信与交互	最小集			扩充集			小结
	可视化的工具集	从设计到实现的转变	对主体开发方法的支持	安装与配置	工具集完备性	知识基础	
ADK	1	0.5	0	0.5	0.5	0.5	3
AgentBuilder	1	1	0.5	1	1	0	4.5
AgentFactory	1	1	1	1	1	0	5
AgentTool	1	0	1	0.5	1	0.5	4
DECAF	1	0.5	0	0	0.5	0.5	2.5
Grasshopper	1	0.5	0	1	1	0.5	4
JACK	1	0.5	0	1	1	0	3.5
JADE	1	0	0	0.5	0.5	1	3
MadKit	1	0.5	0	1	0	0.5	3
Zeus	1	1	0.5	0.5	1	0	4

7. 文档视角 该组的评价结果如表8所示。需要注意的是，对“实例”而言，0分表示没有相关实例，0.5分表示仅有单独的小实例，1分表示实例丰富并且难度循序渐进。

完整、详尽、易于理解的文档对于帮助开发人员快速掌握 MASDE 的用法非常重要。大多数 MASDE(包括 ADK、AgentBuilder、Grasshopper、MadKit、JACK、JADE 和 Zeus)提供了很好的文档的支持，而 AgentFactory、AgentTool 和 DECAF 则不如人意。

表8 文档视角的评价结果

体系结构组	最小集			扩充集	小结
	用户使用指南	用户开发指南	可重用库文档	实例	
ADK	1	1	1	0.5	3.5
AgentBuilder	1	1	1	0.5	3.5

AgentFactory	0.5	0	0	0	0.5
AgentTool	1	0	0	0	1
DECAF	1	1	0	0	2
Grasshopper	1	1	1	1	4
JACK	1	1	1	1	4
JADE	1	1	1	0.5	3.5
MadKit	1	1	1	1	4
Zeus	1	1	1	1	4

8. 生产商视点　该组的评价结果如表9所示满分4分。

表9　生产商视角的评价结果

兼容性组	最小集		扩充集		小结
	不断有新版本和补丁推出	邮件列表	咨询和培训	第三方技术支持	
ADK	1	1	1	0	3
AgentBuilder	0	1	1	0	2
AgentFactory	0	0	0	0	0
AgentTool	1	0	0	0	1
DECAF	1	0	0	0	1
Grasshopper	1	1	1	1	4
JACK	1	1	0	0	2
JADE	1	1	0	1	3
MadKit	1	1	0	1	3
Zeus	0	1	0	0	1

3.2　总体评价

表10给出了一个总体的评分。其中数据的格式为“得分(最小集得分，扩充集得分)”，如第一个数据2(2,0)表示ADK在体系结构组得分为2分，其中最小集2分，扩充集0分。

从MASDE的综合性能(以总分为依据)来看，可以大致将这10个MASDE分为3个等级(从低到高)：(1)一般，包括AgentTool(8.5分)和DECAF(12分)，其中其共同特点为针对某个特定应用研制，比如AgentTool是一个专门支持MaSE开发方法的工具，DECAF是一个所谓的主体操作系统，；(2)中等，包括ADK(22.5分)、AgentBuilder(23分)、JACK(19分)、JADE(23分)和Madkit(18.5分)，其共同的特点为某些方面表现较好；(3)良好，包括AentFactory(27.5分)、Grasshopper(26.5分)和Zeus(26.5分)，其共同的特点为各方面比较均衡，都达到较好的程度。

而MASDE的主要发展趋势为：(1)所支持的单主体体系结构多为几个基本主体体系结构的混合体，而混合式体系结构兼有各个基本体系结构的特点，将为越来越多开发者所青睐。(2)Java语言已经成为主体实现语言的首选，仅有少数MASDE使用专门为面向主体编程(AOP)研制的主体语言。不过，预计面向对象语言的广泛使用仅仅是过渡时期的选择，面向主体编程最终还是需要依靠面向主体的编程语言来实现。(3)所使用的消息传输机制都基于常见的TCP/IP、RMI或CORBA等机制，所使用的主体通信语言主要是KQML和FIPA ACL，表明这两个方面的标准化大致完成，并为主体系统之间的互操作打下了一个好的基础。(4)良好的工具集在主体实现中发挥越来越大的作用，但同时也要提供清晰易懂的文档来帮助用户迅速掌握其使用。

表10　典型MASDE评价总体结果

	体系结构5 (4,1)	通信与交互5 (3,2)	兼容性3 (1,2)	面向主体的软件工程6 (4,2)	主体平台10 (3,7)	工具集6 (3,3)	文档4 (3,1)	开发商4 (2,2)	总分43 (23,20)
ADK	2(2,0)	2(2,0)	0	0	9(3,6)	3(1.5,1.5)	3.5(3,0.5)	3(2,1)	22.5(13.5,9)
AgentBuilder	2(2,0)	4(3,1)	0	5(4,1)	2(2,0)	4.5(2.5,2)	3.5(3,0.5)	2(1,1)	23(19.5,4.5)
AgentFactory	3(3,0)	4(3,1)	0	5.5(4,1.5)	9.5(3,6.5)	5(3,2)	0.5(0.5,0)	0	27.5(16.5,9.5)
AgentTool	0	0	0	2.5(2,0.5)	0	4(2,2)	1(1,0)	1(1,0)	8.5(6,2.5)
DECAF	1(1,0)	1(1,0)	0	0	4.5(3,1.5)	2.5(1.5,1)	2(2,0)	1(1,0)	12(9.5,2.5)
Grasshopper	1(1,0)	2(2,0)	3(2,1)	0	8.5(3,5.5)	4(1.5,2.5)	4(3,1)	4(2,2)	26.5(14.5,12)
JACK	3(2,1)	1(1,0)	0	1.5(1,0.5)	4(3,1)	3.5(1.5,2)	4(3,1)	2(2,0)	19(13.5,5.5)
JADE	2(1,1)	3(2,1)	2(2,0)	0	6.5(3,3.5)	3(1,2)	3.5(3,0.5)	3(2,1)	23(14,9)
MadKit	1(0,1)	1(1,0)	0	0	6.5(3,3.5)	3(1.5,1.5)	4(3,1)	3(2,1)	18.5(10.5,8)
Zeus	2(2,0)	5(2,3)	0	4(4,0)	6.5(3,3.5)	4(2.5,1.5)	4(3,1)	1(1,0)	26.5(17,9)

然而，目前MASDE的发展也有若干问题存在：1)多数MASDE对多主体体系结构(或称组织模型)的支持还不够，应该引起更多开发者的注意。2)尽管面向主体的软件工程是当前研究的热点，多数MASDE仍然缺乏良好的面向主体的开发方法和主体建模语言的支持。3)多数MASDE所提供的主体平台对各种主体所需服务的支持还不够，而这些服务对于提高多主体系统的可重用性、健壮性和可靠性都是很有益的。4)多数MASDE对主体互操作性的支持仍然不够，而互操作性对于提高多主体系统的可重用性、开发性和灵活性是很重要的。

相关工作与结论　目前主要有两类人员关注对MASDE的分析和评估：(1)MASDE的开发商，比如IntelliOne Technologies公司的网站http://www.agentbuilder.com上有一个MASDE列表，(2)某些组织和研究人员，比如Agentlink组织的网站www.agentlink.org上也有一个MASDE列表。文[1]从三个角度(商业因素、技术因素和实践因素)来评价MASDE，但主要强调商业因素，且仅提出了理想MASDE的特征和服务，而没有对任何MASDE作出具体的分析和评估。文[2]主要从面向主体软件工程的角度来评价MASDE，把软件开发过程分为4个阶段(分析、设计、开发和部署)，每个阶段使用一组性质来衡量，并以此为基础分析和比较了4个MASDE。文[3]从5个方面(主体属性、软件工程支持、主体和多主体系统实现、技术问题、经济因素)提出了一套较为详细的标准，并以此为基础对5个MASDE做了一个较为详细的分析和评价。但该项工作的主要目的不在于制定这套标准，而在于提出根据应用特征选择MASDE的若干方针。文[4]提出了一个14条的标准集，并采用定量的方式对8个MASDE进行了综合分析和比较，但其标准集繁杂，缺乏条理。

分析上述各种MASDE评估方法，主要存在如下问题：(1)这些工作[1~4]都没有把评价标准集的制定作为研究工作的重点和核心。而本文认为，制定一个多视角、易操作的评

(下转第242页)

在最坏情况下的计算时间为 $O(3(n-1)$，而 while 循环体在最坏情况下要执行 $2n-1$ 次，故最坏情况下函数 *fenzhi* 所需的时间复杂度为 $O(3(n-1)*(2n-1))$，故算法最坏情况下的时间复杂度为 $O(n^2)$。

5 两种算法的简单比较

(1) 尽管上面两种算法的最坏时间复杂度均为 $O(n^2)$，但回溯算法在搜索过程中即使找到了最优解也不能马上确定它就是最优的，只有把它所有的可行解全部求出后才能确定最后的最优解，而分支限界法则只要搜索到一个解，这个解必为最优的，所以总体而言采用分支限界法将会比回溯法检查更少的结点，算法的平均时间要比回溯法节省；

(2) 上述两种算法中，回溯法占用的内存是解空间的最大路径长度个单元，而分支限界法所占用的内存为整个解空间的大小，对于一个子集空间，回溯法通常需要 $O(n)$ 的内存空间，而分支限界法则需要 $O(2^n)$ 的空间；对于排列空间，回溯需要 $O(n)$ 的内存空间，分支限界法界需要 $O(n!)$ 的空间。

6 实验结果

当 $P_{max}=200$，$P_{min}=80$，$d=60$ 时，我们针对图2所示的网络图对两种算法进行了实验，输出结果如图3所示，其中 a 图是建立存储结构的输入界面，b 图是算法的输出结果。由于采用了相同的输出界面，所以两种算法的输出结果是相同的。

结束语 本文针对石油传输网络最少增压器问题分别采用了回溯法和分支限界法进行了算法设计，并对这两种算法的时间和空间复杂度进行了分析和比较，实验结果验证了算法的有效性。

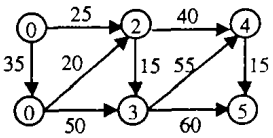


图2 石油传输网络图示

```
请输入6个顶点                    请输入源点 S 处的油压值
0 1 2 3 4 5                      s=120
请输入9条边——i,j,w               结果为
0 1 35                           best×[0]=1
0 2 25                           best×[1]=1
1 2 20                           best×[2]=0
1 3 50                           best×[3]=0
2 3 15                           best×[4]=0
2 4 40                           best×[5]=0
3 4 55
3 5 60
4 5 10
```

a　　b

图3 算法试验结果

参 考 文 献

1 苏士峰. 中国油气管道在创新中发展. http://www.gdb.net.cn/share.asp,2004

2 王晓东. 计算机算法设计与分析. 北京：电子工业出版社，2001

3 王岩冰，郑明春，刘弘. 回溯算法的形式模型. 计算机研究与发展，2001，38(9)：1066～1079

4 backtracking. http://www.ibrtses.com/delphi/backtracking.html,2003

5 分支限界问题分析. http://www.frontfree.net/articles/services/view.asp?id=669&page=1,2002,11

(上接第205页)

价标准集是整个评价工作的基础和关键。(2)部分工作[3,4]给出了一个内容庞杂、条目众多的标准集，却没有提供一种有效的、结构化方式来组织这些标准集，因而显得结构混乱而内容不清晰。(3)多数工作[3,4]所提出的评价标准只是一个抽象的概念，而不是一种具体机制，很难对是否符合进行判断，因而可操作性差，不易使用。

与已有工作相比较，本文提出的 SEF 具有多视角、易于操作、可扩展以及独立于应用等特点，能够更好地用于描述、分析和展示 MASDE 的基本特征，发现其优势和不足。SEF 由多个组构成，一个组代表了一个不同的视角，因此是多视角的；每个评价条目是小粒度的、底层的、具体的，因而是易于操作的；每个组的评价条目分为最小集和扩展集，扩展集可以根据需要进行扩展，因而是可扩展的；最后，所有评价条目的选择都是独立于具体应用的。综合来说，SEF 是一个完整、有效的 MASDE 的结构化评价框架。

基于 SEF，本文进一步对一组有代表性的 MASDE 进行了深入的剖析，得到了一系列重要的评价细节和结果，并具有以下特点：(1)覆盖面广，本文总共选择了十个典型的 MASDE 进行评价，基本上覆盖了各种类型的 MASDE。(2)具有分类评价结果和总体评价结果。

参 考 文 献

1 Schoepke S H. A Business View Regrading the Selection Of Agent Development Tookits. In: Proc. of the AAAI-98 Workshop on Software Tools for Developing Agents,1998

2 Ricordel P M,Demazeau Y. From analysis to deployment: a multi-agent platform survey. In:Proc. of the Workshop on Engineering Societies in the Agents' World,Springer-Verlag,2000

3 Eiter T, Mascardi V. Comparing Environments for Developing Software Agents, AI Communications,2002,15(4)

4 Garneau T,Delisle S. A New General, Flexible and Java-Based Software Development Tool for Multiagent Systems. ISE 2003, 2003

5 FIPA. FIPA Abstract Architecture Specification (FIPA0001). Available at http://www.fipa.org,2000

6 FIPA. FIPA Agent Management Specification (FIPA00023). Available at http://www.fipa.org, 2000

7 OMG. OMG Mobile Agent System Interoperability Facility (MASIF). Available at http://www.omg.org,1997

8 Tryllian. The Developer's Guide(ADK 2.1 version). Available at http://www.tryllian.com,2002

9 Regular Systems. AgentBuilder User's Guide Version 1.3. available at http://www.agentbuilder.com, 2000

10 Collier R W. Agent Factory: A Framework for the Engineering of Agent-Oriented Applications:[Ph. D. Thesis]. University College Dublin, Ireland, 2001

11 DeLoach S A. Analysis and Design using MaSE and agentTool. In:12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2001), April 2001

12 Graham J R,Decker K S,Mersic M. DECAF - A Flexible Multi Agent System Architecture. Appearing in Autonomous Agents and Multi-Agent Systems,2003

13 Baumer C,Breugst M,Choy S,Magedanz T. Grasshopper--a universal agent platform based on OMG MASIF and FIPA standards. In:First Intl. Workshop on Mobile Agents for Telecommunication Applications (MATA'99), Oct. 1999

14 Busetta P,Ronnquist R, Hodgson A, Lucas A. Jack Intelligent Agents - Components for Intelligent Agents in Java. AgentLink News Letter, Janvier 1999

15 Bellifemine F, Poggi A, Rimassa G. JADE-A FIPA2000 Compliant Agent Development Environment. AGENTS'01,2001

16 Ferber J,Gutknecht O. A meta-model for the analysis and design of organizations in multiagent systems. In: Third Intl. Conf. on Multi-Agent Systems (ICMAS '98), IEEE Computer Society, 1998

17 Nwana, Ndumu,Lee,Collis. ZEUS: A Tool-Kit for Building Distributed Multi-Agent Systems. Applied Artifical Intelligence Journal, 1999