

自然约束语言

The NCL Natural Constraint Language

周建阳 © 著



科学出版社
www.sciencep.com

自然约束语言

The NCL Natural Constraint Language

周建阳 著

科学出版社

北 京

内 容 简 介

本书介绍自然约束语言 NCL 及其开发平台 POEM。全书共 6 章, 其中第 1 章简要介绍 NCL 语言与求解系统; 第 2 章和第 3 章介绍 NCL 语言的基本体系和原理方法, 内容包括 NCL 的词法、语法及语义等; 第 4 章介绍 NCL 语言的开发平台 POEM 的使用方法; 第 5 章介绍如何用 NCL 语言进行建模及求解; 第 6 章介绍 NCL 语言在工业优化中的部分应用。

本书可作为高等院校及科研院所研究运筹学、物流优化、人工智能和软件方向的教师和研究生的科研参考书。针对如何用 NCL 语言及其开发平台 POEM 求解运筹学组合优化问题, 本书可作为企事业单位中从事生产制造、物流信息化、人力资源优化等工作的 IT 人员研究计划、排程与优化的指导书, 也可以作为 POEM 软件平台配套的参考手册。

图书在版编目(CIP)数据

自然约束语言 = The NCL Natural Constraint Language/周建阳著. —北京: 科学出版社, 2009

ISBN 978-7-03-024973-9

I. 自… II. 周… III. 运筹学 IV. O22

中国版本图书馆 CIP 数据核字 (2009) 第 115854 号

责任编辑: 王丽平 房 阳 / 责任校对: 陈丽珠

责任印制: 钱玉芬 / 封面设计: 王 浩

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

源海印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2009 年 7 月第 一 版 开本: B5 (720 × 1000)

2009 年 7 月第一次印刷 印张: 15 1/2

印数: 1—3 500 字数: 297 000

定价: 48.00 元

(如有印装质量问题, 我社负责调换〈环伟〉)

前 言

设计一门接近人类的基础推理、以常规数理逻辑为语法的问题求解语言是可行的,这一目标可以通过结合人工智能(Artificial Intelligence)、运筹学(Operations Research)和逻辑规划(Logic Programming)技术来实现。本书为读者介绍自然约束语言(Natural Constraint Language, NCL)及其软件平台 POEM[®] (Programming in Operation and Expressive Models)。NCL 使问题建模及求解近乎自然,提供给用户一门学习快捷、使用简便、以自然简洁的方法求解组合问题的计算机语言。

作者在 1995 年博士学习阶段开始构思研发 NCL,设计思想相当简单,使用 T_EX 作为元语言来描述组合问题并用 NCL 解算器进行问题求解。简而言之,NCL 是一门支持智能语法(上下文相关语法)的、求解约束满足问题的描述型语言,支持隐式类型的声明、进行全局的语义分析、基于上下文的推理及求解。NCL 将数值约束、简化的一阶逻辑及集合推理集成在一个语言环境之下,形成一个在混合域(实数、整数、布尔值、索引及集合)上针对约束满足问题的联合求解系统。因此,NCL 具有独特的编程风格:① 自然建模;② 混合集合规划。

NCL 的语言原型于 1997 年 12 月提交给第三届国际系统科学和系统工程会议(Zhou, 1998),并于 1998 年 3 月正式提交给逻辑规划协会的官方杂志 *The Journal of Logic Programming* (Zhou, 2000)。

迄今为止,本书介绍的 NCL 距 NCL 的原型已十年有余。NCL 从最初几万行的 C++代码发展到现今的 30 余万行,技术趋于成熟。产品 POEM 的总体代码量也已突破 70 万行的 C++代码,成为一个大型的、支持工程化开发优化方案的基础型软件。NCL 朝着“将数理逻辑产业化”的目标前进了一步。

作 者

2009 年 3 月 16 日

目 录

前言

第 1 章	NCL 与求解系统	1
1.1	求解系统	1
	解算器(SOLVER)	1
	语法分析器(PARSER)	2
	规则(RULES)	3
1.2	NCL 语言简介	3
	自然建模(NATURAL MODELING)	3
	混合集合规划(MIXED SET PROGRAMMING)	3
	求解规则(SEARCH RULES)	4
	NCL 是联合求解系统	4
1.3	基于 NCL 的 POEM [®] 平台	5
第 2 章	NCL 的词法	7
2.1	常规词法	7
	字符	7
	标识符	7
	特殊标识符	7
	常量	8
	未确定值	8
	注释	9
2.2	数学编码	9
	数学符号一览表	9
	函数一览表	10
	T _E X 聚合符	11
2.3	数据类型	12
	广义数据类型	12
	逻辑推理的数据精度	13
	集合类型	13
	日期/时间类型	14
	缺省值	14
	数据示例	14
第 3 章	NCL 的语法及语义	16
3.1	NCL 的常规逻辑	16

语句(STATEMENT).....	16
量词(QUANTIFICATION).....	17
索引(INDEX).....	18
条件句(CONDITIONAL).....	19
约束(CONSTRAINT).....	20
表达式(EXPRESSION).....	22
浮点数表达式(FLOAT EXPRESSION).....	22
整数表达式(INTEGER EXPRESSION).....	22
字符串(STRING).....	23
逻辑指针(REFERENCE).....	23
集合表达式(SET EXPRESSION).....	24
布尔表达式(BOOLEAN EXPRESSION).....	24
聚合式(AGGREGATION).....	24
常量(CONSTANT).....	28
输入/输出的格式(INPUT AND OUTPUT FORMAT).....	29
日期/时间格式(DATE/TIME FORMAT).....	29
日期/时间的属性函数(DATE/TIME ATTRIBUTE).....	30
变量(VARIABLE).....	31
匿名变量(ANONYMOUS VARIABLE).....	31
连缀(CONCATENATION).....	31
个性化消息(CUSTOM MESSAGE).....	32
软约束(SOFT CONSTRAINT).....	32
数据源(DATA POOL).....	33
输入/输出的指定(I/O SPECIFICATION).....	33
输出(PRINT).....	34
宏调用(INCLUDE).....	34
3.2 NCL 的时态逻辑.....	34
系统变量(SYSTEM VARIABLE).....	35
抽取(EXTRACTION).....	37
赋值(ASSIGNMENT).....	38
跳转(GOTO).....	38
子模型(SUB MODEL).....	39
SQL 语言接口.....	40
操作系统的 OS 命令(OS COMMAND).....	41
期待约束(EXPECTATION CONSTRAINT).....	41
3.3 NCL 的求解逻辑.....	41
切削与搜索(CUT AND SEARCH).....	41

查询与搜索(QUERY AND SEARCH)	42
枚举方式(ENUMERATION MODE)	42
查询准则(QUERY CRITERIA)	43
优化目标(OPTIMIZATION OBJECTIVE)	44
求解过程的示范	44
对求解的系统控制	46
3.4 消息与跟踪管理	47
NCL 消息(NCL MESSAGE)	47
终止状态(TERMINATION STATUS)	49
可编程的暂停(PROGRAMMED BREAK)	49
可编程的调试(PROGRAMMED DEBUGGING)	49
消息处理器(MESSAGE HANDLER)	50
3.5 NCL 的语法范例	51
布尔逻辑(BOOLEAN LOGIC)	51
无穷大(INFINITY)	51
数值约束(NUMERIC CONSTRAINTS)	52
集合推理(SET REASONING)	53
量词(QUANTIFICATION)	54
混合集合规划示例(MIXED SET PROGRAM)	57
分支(SWITCH)	58
规则(RULE)	60
优化目标(OPTIMIZATION OBJECTIVE)	60
输入/输出机制(I/O FACILITIES)	61
查询及搜索(QUERY AND SEARCH)	62
跳转(GOTO)	63
聚合(AGGREGATION)	64
用作下标的指针(REFERENCE SUBSCRIPTS)	67
被引用的运算式(REFERENCED OPERATORS)	69
连缀(CONCATENATION)	70
日期/时间的管理(DATE/TIME MANAGEMENT)	72
抽取及时态逻辑(EXTRACTION AND TEMPORAL LOGIC)	75
赋值(ASSIGNMENT)	76
子字符串及集合的元素(SUBSTRING AND ELEMENTS FROM SET)	77
获取集合的分段区间(OBTAINING PIECEWISE INTERVALS FROM A SET)	79
个性化消息(CUSTOM MESSAGE)	80
内存缓冲区数据源(BUFFER POOL)	80
存储于文件的子模型(SUB MODEL IN A FILE)	81

存储于内存缓冲区的子模型(SUB MODEL IN A BUFFER POOL)	82
子模型的返回值(RETURN VALUES OF A SUB MODEL)	83
嵌套调用(NESTED CALL)	83
子模型调用溢出(OVERFLOW IN A SUB MODEL CALL)	84
数据库连接及 SQL 查询(DATABASE CONNECTION AND SQL QUERY)	84
操作系统 OS 命令(OS COMMAND)	87
匿名变量及缺省值(ANONYMOUS VARIABLES AND DEFAULT VALUES)	88
系统变量(SYSTEM VARIABLE)	90
软约束(SOFT CONSTRAINT)	91
用期待约束进行程序调试(PROGRAMMED DEBUGGING WITH EXPECTATION CONSTRAINT)	91
第 4 章 NCL 语言的开发平台 POEM®	93
4.1 POEM 的主界面	93
工具栏(TOOL BAR)	94
T _E X 符号栏	94
工作区(WORKSPACE)	95
编辑窗(EDIT WINDOW)	96
跟踪窗(TRACE WINDOW)	96
4.2 项目配置	97
NCL 的数据源	97
NCL 的参数配置	97
项目配置窗	101
4.3 模型夹及模型库	102
模型夹(MODEL FOLDER)	102
NCL 模型库(MODEL LIBRARY)	103
4.4 信息表	103
现行模型表(RUNNING MODELS)	104
常量表(CONSTANTS)	104
变量表(VARIABLES)	105
约束表(CONSTRAINTS)	106
4.5 视图及调试	107
快捷查视(QUICK WATCH)	107
浏览器(BROWSER)	108
约束调试器(CONSTRAINT DEBUGGER)	109
可视化调试器(VISUAL DEBUGGER)	109
结果可视化窗(SOLUTION VIEWER)	110
4.6 跟踪窗与工作模式	111

调试模式(DEBUG MODE).....	112
计时模式(TIMER MODE).....	113
跟踪级别(TRACE LEVEL).....	113
诊断信息的选项窗.....	113
统计信息的选项窗.....	114
推荐的模型诊断模式.....	114
推荐的正常工作模式.....	114
4.7 在线帮助.....	115
第5章 建模及求解.....	117
5.1 工程化建模.....	117
建模步骤.....	117
变量的命名公约.....	117
主动式模型改进.....	119
被动式模型改进.....	120
5.2 NCL 的模型抽象.....	120
两两不等的整数(DISTINCT INTEGERS).....	120
两两不交的集合(DISJOINT SETS).....	121
排序(SORTING).....	121
集合的覆盖与划分(SET COVERING AND PARTITIONING).....	123
拼排(PACKING).....	125
有限能力(FINITE CAPACITY).....	125
求和(SUM).....	126
二维累积(CUMULATION).....	126
5.3 智力游戏(PUZZLES).....	127
字谜(SEND MORE MONEY).....	127
素数问题(PRIMES).....	128
整数排序(INTEGER SORTING).....	128
皇后问题(QUEENS).....	129
神奇的方块(MAGIC SQUARE).....	130
数独(SUDOKU).....	132
神奇的序列(MAGIC SEQUENCE).....	134
爱因斯坦的游戏题(EINSTEIN'S QUIZ).....	135
数谜(CALCULS D'ENFER).....	136
方块拼排(SQUARE PACKING).....	138
骑士问题(KNIGHT).....	139
5.4 求解复杂问题.....	142
集合划分(SET PARTITIONING).....	142

高尔夫球对抗赛(GOLF TOURNAMENT)	143
赛舟会(PROGRESSIVE PARTY)	145
货船装载(SHIP LOADING)	149
车间排序(JOB-SHOP SCHEDULING)	152
最小化热能转换器的能耗(MINIMIZING THE COST OF A HEAT EXCHANGER)	156
带时间窗的取货与送货(PICKUP AND DELIVERY WITH TIME WINDOWS)	157
练习题	165
5.5 松弛逻辑与二次优化	165
交互逻辑(INTERACTION LOGIC)	165
迭代优化(ITERATIVE OPTIMIZATION)	165
旅行商问题的迭代优化方法(ITERATIVE OPTIMIZATION FOR TSP)	166
TSP 的练习题	172
第 6 章 NCL 的工业应用	173
6.1 生产排程	173
问题定义	173
数据逻辑	175
简化的优化模型	179
时间的可视化工具: 甘特图(GANTT CHART)	184
练习题	184
6.2 人员排班计划	184
问题定义	184
数据逻辑	185
简化的优化模型	190
统计信息的可视化工具: 直方图(HISTOGRAM)	196
练习题	197
6.3 多式联运优化	197
问题定义	197
数据逻辑	197
简化的优化模型	202
地理信息的可视化工具: 地图(MAP)	210
练习题	210
参考文献	211
附录 1 NCL 语法的 T_EX 编码	213
附录 2 ComPoem ActiveX 组件	227
英文索引	230
中文索引	233

第 1 章 NCL 与求解系统

1.1 求解系统

约束满足问题(Constraint Satisfaction Problem)在日常生活与工作中无处不在,很多都属于 NP 困难(NP-hard)型。复杂性理论(Complexity Theory)表明,除非 P 类问题等于 NP 类问题,一个问题如果是 NP 完备型(或 NP 困难型)则意味着不存在求解此问题的多项式时间的算法(Lenstra and Kan, 1979)。

本书着重讨论针对约束满足问题的求解系统的三项关键技术:语法分析器(Parser)、解算器(Solver)、规则(Rules)。之所以论述这三项技术,是因为它们分别涉及数学建模、解算及对求解的规范。以下先介绍求解系统最核心的解算器,再论述语法分析器及规则。

解算器(SOLVER)

解算器是求解系统的核心,一方面它是一个算法引擎,另一方面它是一个推理系统。本书着重介绍逻辑化、工业化的求解系统。

运筹学与线性规划

运筹学是系统研究经济、军事等活动中有关决策、管理的问题的一门科学。提到运筹学,就不免提到线性规划(Linear Programming)——求解以线性函数为优化目标的线性约束系统的技术。George B. Dantzig 在 1947 年发明高效的单纯形算法(Simplex)解线性规划问题(Dantzig, 1990)之后,运筹学界一步步建立起基于线性松弛算法的数学规划(Mathematical Programming)体系:线性规划(Linear Programming)、整数线性规划(Integer Linear Programming)、0-1 规划(0-1 Programming)、混合整数规划(Mixed Integer Programming)……

逻辑规划

逻辑规划(Logic Programming)是一门基于数理逻辑(Mathematical Logic)的计算机编程的技术。提到逻辑规划,就不免要提到 Prolog 语言。马赛大学的 Alain Colmerauer 等发明了基于谓词逻辑的逻辑语言 Prolog(Colmerauer, 1973),使程序员在一种崭新的视角下进行编程,声明型语言(Declarative Language)的概念也开始更加清晰地区别于过程型语言(Procedural Language)。

约束规划

约束规划(Constraint Programming)是以约束形式描述变量间关系的规划框架(Colmerauer, 1990; Dincbas et al., 1988; Jaffar and Lassez, 1987), 它的发展已有 20 多年的历史。约束规划的思想是将求解系统置于逻辑规划的框架下来解问题, 其最大意义在于它开拓了一个新的求解约束满足问题的方法框架。有别于传统的数学规划方法, 约束规划打破了线性建模的限制, 从通用模型的角度去研究问题求解的算法, 但是到目前为止, 这一规划框架的发展历史相比数学规划还属“稚嫩”。在集合划分、路径优化等问题领域, 约束规划技术尚未有令人满意的结果。

混合集合规划

线性规划解线性问题非常高效, 但它有如下的局限性:

- 概念的局限性 求解系统基于单纯形等线性松弛算法, 并以线性模型为框架;
- 方法的局限性 求解非线性问题需作线性转化, 难于处理复杂的边际约束;
- 应用的局限性 难于求解带复杂边际约束的非线性问题, 难于应对复杂的运行环境。

约束规划将求解系统逻辑化, 并打破了线性建模的限制, 具有很高的灵活度, 但在求解性能上遇到了很大的困难。

基于以上观察, 本书论述在混合域(实数、整数、布尔、集合)上的、基于集合规划的算法体系, 作者称之为混合集合规划(Mixed Set Programming, MSP)(Zhou, 2008), 是以一阶逻辑与集合推理(Halmos, 1960)为算法内核的逻辑求解系统。混合集合规划源于约束规划。MSP 是本书讨论的主要算法体系, 在随后章节中将系统展开。

语法分析器(PARSER)

基于 C++或 Java 等语言的算法库并不能让用户在高层级上求解问题, 而现实中的工业问题往往极端复杂且规模巨大。例如, 实际的生产排程问题往往涉及数十类的约束, 达到数万道工序的量级, 并对实时性、直觉式交互、二次优化等有很强的要求。解决方案从研发到实施一般很困难, 因为研发成本高、部署周期长、见效慢。为有效实施大规模复杂问题的优化方案, 就需要采用平台化的开发工具进行工程化的开发。平台化的开发工具依赖高层级的语言系统, 这就彰显出智能语法分析器的重要性。

智能的语法分析一般意义上指在语义层面进行语法分析。不同于如 C++及 Java 等过程型语言采用的上下文无关的语法(Context Free Syntax), 智能的语法是上下文敏感的(Context Sensitive)。智能的语法分析器使优化方案开发者不必拘泥于算法细节, 可以精确、高度简洁、概念清晰地对实际问题进行描述建模。

规则(RULES)

NP 困难问题是对算法的巨大挑战(Computational Challenge), 因为几乎可以不抱任何的侥幸心理, 没有任何精确型算法可确保对该类问题的求解, 无论多么强大的解算器都是“脆弱”的。在这样的前提下, 将规则引入求解系统是自然的, 一方面, 业务规则(Business Rules)将领域的或行业的业务逻辑引入求解过程, 帮助系统快速地、个性化地构建合乎业务逻辑的解; 另一方面, 规则在求解过程中对搜索解空间起到规范作用, 策略性地抑制组合爆炸。

规则系统可以独立成为一个求解系统, 但将规则引入以精确的解算器为核心的求解系统则使求解性能更为完善。

1.2 NCL 语言简介

NCL 语言是一门以标准的数理逻辑为语法的描述型语言(Zhou, 2000)。在这一意义上, NCL 语言是新一代的语言, 与传统声明型语言及脚本型建模语言不同, NCL 是第一个采用上下文相关语法的、用于求解约束满足问题的描述型语言。因而, NCL 支持对变量类型的隐式声明, 可根据上下文约束进行推理及求解。

简而言之, 不同于常规约束及逻辑语言, NCL 的科技创新可概括为如下三方面:

- 自然建模(Natural Modeling);
- 混合集合规划(Mixed Set Programming);
- 求解规则(Search Rules)。

自然建模(NATURAL MODELING)

NCL 支持以常规数理逻辑为语法的自然建模, 人工智能(Artificial Intelligence)的模式识别技术广泛应用于 NCL 的语法分析、语义识别、模型诊断及模型优化上, 使基于 NCL 的编程文档化。文档化的程序支持简洁、清晰的业务逻辑, 使编程者可将主要精力用于高层级的、业务逻辑一级的模型设计, 而无需在低层级、算法细节上停滞。

混合集合规划(MIXED SET PROGRAMMING)

本书的集合规划的概念并非指在问题求解中对集合标注(Set Notation)、集合变量(Set Variable)及集合约束(Set Constraint)的简单使用。本书的集合规划是指系统地将集合推理与运筹学算法相结合, 以集合变量为主进行问题建模, 以基于集合推理的算法为主求解问题。读者在阅读完本书(特别是 NCL 的工业应用)后可以看到, 几乎对所有的复杂问题的建模与求解均是以基于集合推理的算法模型为主的。

MSP 是指: ① 实数、整数、布尔值、索引及集合类型的混合域上的全局推理; ② 将简化的一阶逻辑、集合推理、数值约束及运筹学算法集成于一个语言系统, 用于处理约束满足问题。NCL 的混合集合规划技术的目标为实现一种能够超越线性限制的、替代(基于线性松弛算法的)线性解算器的、更通用的算法系统来求解约束满足问题, 有效处理集合覆盖/集合划分、生产排程及路径优化等问题。

在作者所知范围内, NCL 是迄今唯一具有混合集合规划算法体系的语言系统。

求解规则(SEARCH RULES)

支持基于规则的求解, 使用户可对求解规则高度简洁地进行编程。一方面, 在不同的业务领域存在经验性的业务规则, 可将业务规则用于 NCL 的求解规则以规范在搜索树上的求解; 另一方面, 运筹学的研究衍生出丰富的启发式求解规则(如最小松弛度、最小遗憾度、贪婪性搜索等), 可将启发式求解规则用于 NCL 的求解规则以规避组合爆炸。

NCL 是联合求解系统

可见, NCL 是一个联合求解系统, 其求解框架的设计考虑到求解系统的构成要素: 建模、解算及规范求解, 其思想是将语法分析器、解算器、规则三项技术集成于一体。Parser 与 Solver 相结合, 使求解系统可在高层级上用简洁的数学模型自然清晰地定义问题; Rules 与 Solver 相结合, 使求解系统可用业务逻辑快速地、个性化地规范求解。

简而言之, NCL 高度整合人工智能(AI)、运筹学(OR)及逻辑规划(LP)于一体, 成功地将运筹学(OR)算法封装于简洁的逻辑界面下。不同于大多数的逻辑及约束系统, 全局约束(Global Constraint)的概念不适用于 NCL。事实上, NCL 内核包含数百条基础的及全局的约束。全局约束在问题模型层面上是好的概念, 但如果要将全局约束在 NCL 中显式定义, 则必须引入非常多的特殊命名、命名的变种及外围概念, 同时在使用手册上也必须一一给予解释。比如, NCL 包含几十个诸如排序(Sort)的全局算法, 单单 Sort 一个算法就涉及数值或集合、三组变量或二组变量的多种排序算法, 相关的外围概念有 UnaryResource, AscendingSorting 及 RegretBasedSearch 等。如果 NCL 也显式定义全局约束, 那么用户将会被淹没在成百上千个特殊概念中, 用 NCL 编程将会非常的笨拙。为避免出现这样的窘境, NCL 采用智能语法分析器(Semantic Parser)识别约束满足问题的自然建模, 并自动引导 NCL 解算器智能的求解问题。Semantic Parser 技术将编程者从采用如 Sort 一类的预定义谓词来进行特定建模的困境中解脱出来。

总体而言, NCL 亦可被视为约束逻辑语言, 它将简化的一阶逻辑(布尔逻辑、量词逻辑等)、数值约束(整数、浮点数等)、引用、集合推理及实数域(有限精度)的数值分析集成于一体。在词法上, NCL 严格遵从数理逻辑的符号标准, 采用

T_EX(Knuth, 1984)的数学编码清晰表述。在语法与语义上, NCL 支持上下文相关的语法分析、集合规划、动态的约束求解和模糊推理。在约束处理能力上, 动态量词、引用、抽取、元级连缀、分支、规则、跳转、查询及多目标优化等逻辑与元级控制被自然地引入 NCL。进一步而言, SQL 语言、宏文件调用、子模型及 OS 命令的嵌入使 NCL 成为一门实用的语言, 可进行数据处理、系统集成及模块化编程。

1.3 基于 NCL 的 POEM[®]平台

设计 POEM[®]的目的是为用户提供一个可高效建模及求解大规模组合问题的编程环境。

在编程语言意义上, POEM 是 NCL 的高效实现。在实用系统意义上, POEM 整合 NCL、项目管理器(Project Manager)、信息表(Information Tables)、调试器(Debugger)、消息处理器(Message Handler)及结果可视化窗(Solution Viewer)。丰富的功能模块使 POEM 成为针对计划与排程应用进行快速的开发、调试、测试、建文档、部署与维护的一个完整平台。

除核心语言及主界面外, POEM 包括如下的外围模块:

- 项目管理器(Project Manager)
- 模型夹(Model Folder)
- 信息表 现行模型表(Running Models)、消息表(Messages)、常量表(Constants)、变量表(Variables)、约束表(Constraints)
- 调试器 快捷查视(Quick Watch)、浏览器(Browser)、约束调试器(Constraint Debugger)、可视化调试器(Visual Debugger)
- 结果可视化窗(Solution Viewer)
- 消息处理器(Message Handler)

.....

POEM 外围模块的设计遵从软件标准。如果用户已经是 Visual C++或 Visual Basic 编程者, 他将很容易掌握 POEM。

通过 POEM 提供的项目管理器(Project Manager), 用户可在工作区窗口同时管理多个项目。更为方便的是, 用户可通过模型夹(Model Folder) 遍历浏览项目目录。

在高层级上, POEM 通过信息表、消息及调试器与用户交流信息。对每个项目, 用户可借助 POEM 的信息表查询运行中模型的结构、系统信息、NCL 常量、变量及约束。

对于每个运行中的项目, 编程者可通过 POEM 调试器动态地跟踪 NCL 模型的运行过程, 快捷查视(Quick Watch)和浏览器(Browser)允许用户查看或浏览 NCL 的表达式及其取值。约束调试器(Constraint Debugger)允许用户随时跟踪约束失败等

运行状况。注意, NCL 消息起着协调用户与 POEM 的交互的作用。更为方便的是, 可视化调试器(Visual Debugger)允许用户以直觉式方式跟踪问题的求解, 得到解后, 通过结果可视化窗(Solution Viewer)对结果进行观察分析。

在低层级上, POEM 消息处理器(Message Handler)提供编程者针对 NCL 消息的事件处理接口。通过开放式消息处理器的过程/函数接口, 用户可在其他程序语言(Visual Basic, Visual C++, Visual J++, C#、Java 及 Asp.net 等)和 NCL 引擎之间设计处理消息的界面。

POEM 软件平台由 ENGINEST 公司(www.enginest.cn)商业化。

第 2 章 NCL 的词法

2.1 常 规 词 法

NCL 词法常规类似于 C 语言，主要区别在于 NCL 关键字以 T_EX 作为元语言来编码。有关 T_EX，参见(Knuth, 1984)。

字符

字母

a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

数字

0 1 2 3 4 5 6 7 8 9

NCL 可见字符

| & ! = < > + - * / . , () [] @ ' " ? ; : # `

特殊输出字符

'\t': 制表符

'\n': 回车符

NCL 不可见字符

举例:

~ é è ç à " £ \$ ¢ ù µ § ²

NCL 特别用□字符表示空格或空。

标识符

规定所有标识符以英文字母开头，由字母、数字或下划线_组成，如

abc12 VAR A101 var_1

特殊标识符

由单引号 ' 引用的标志(Token)，如

'()	'@'	'.'
'Backtrack Count'	'dd/mm/yyyy hh:mm:ss'	
'Search Depth'	'Solution Count'	