第5卷第1期
1994年1月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 5, No. 1
Jan. 1994

分布式实时系统UECnet的描述工具——配置说明语言UECspec*

舒 敏 龚天富

(电子科技大学计算机系,成都 610054)

摘要 大型、复杂的分布式实时系统除了有严格的时间要求外,还需要有更大的灵活性、可预测性和可靠性.本文提出一个分布式实时系统的描述工具——配置说明语言UECspec,用来描述分布式实时系统UECnet应用层的逻辑网结构、进程间的通信链接关系,以及各进程的实时特性.UECnet系统根据UECspec描述自动完成对应用系统的配置分析,初始分配及重配置.

关键词 分布式系统,程序设计语言和软件,并发程序设计,实时,配置.

随着分布式实时系统朝着大型、复杂的方向发展,系统除了有严格的时间要求外,还需要有较大的灵活性、可预测性和可靠性,因此迫切需要研究开发分布式实时应用系统的构造工具.UECspec语言就是为这个目的而设计的,它为用户提供描述分布式实时应用系统配置的手段,以便UECnet[1]系统能静态地对时间资源进行分析,实施自动的初始分配;如果改变配置说明,该系统可以重新配置,以实现系统重构.

UECnet系统分为两层.一层是操作系统核心层,提供分布进程的实时同步与互斥的消息通信机制和对实时事件进行有效处理的进程调度算法.另一层是配置系统层,以配置语言UECspec为基础,提供一个通用的分布式实时任务构造环境.在UECnet系统中,用程序设计语言编写并能完成一定功能的顺序代码段称为模块.进程是模块的实例.该系统中有三类进程:实时进程是对实时事件的最好抽象,它包括周期进程和突发进程;报警进程用于处理异常事件;后台进程用于描述非实时事件.实时进程是一组固定的任务序列,任务的划分是从程序代码中的接收或发送原语起,到下一个接收或发送原语前止.任务具有确定的时间特性,它由三个时间量来刻划:执行时间,截止时间和周期.进程间采用消息传递方式进行通信,进程通信端口需要静态连接.UECnet系统由一组并发进程组合而成.将一组必须运行在同一物理结点上的进程组合在一起构成逻辑结点,一个或多个逻辑结点可以映射到同一物理结点上.

* 本文1991年2月4日收到,1991年7月29日定稿

本课题得到"七·五"预研项目8B—3—3资助.作者舒敏,女,31岁,讲师,主要研究领域为分布式实时系统,分布式实时语言及其环境.龚天富,56岁,教授,主要研究领域为系统软件.

本文通讯联系人:舒敏,成都610054,电子科技大学806教研室

1 UECspec 语言

UECspec 语言的主要功能是描述 UECnet 的逻辑网络的完整结构,以及各逻辑结点中进程的实时特性.它包括如下成分:

(1)说明每个逻辑结点以及它们到物理结点的映射关系.

(2)说明每个模块,包括使用的语言、所在文件名、通信端口等.

(3)说明每个模块实例(进程),包括进程的实时任务、实时特性等.

(4)描述各进程间通信端口的连接关系.

(5)描述系统的配置改变.

1.1 系统

使用本语言的一个完整描述称为一个系统.一个系统由逻辑结点网、模块说明、进程说明和端口连接四部分组成.

```
〈系统〉::=system 〈系统名〉
                 〈逻辑结点网〉
                 〈模块说明〉
                 〈进程说明〉
                 〈端口连接〉
          endsystem
```

1.2 逻辑结点网

```
〈逻辑结点网〉::=〈逻辑结点〉{〈逻辑结点〉}
〈逻辑结点〉::=lnode 〈逻辑结点〉
                proc_num 〈进程个数〉〈进程名〉{〈进程名〉}
                pnode〈物理结点号〉[,〈约束标志〉]
〈逻辑结点号〉::=〈整数〉
〈进程个数〉::=〈整数〉
〈物理结点号〉::=〈整数〉
〈约束标志〉::=1|0
```

逻辑结点网由若干个(至少一个)逻辑结点组成,每个逻辑结点由若干个(至少一个)进程组成,这些进程必须作为一个整体同时分配在一个物理结点上.物理结点号给出了程序员所期望分配的物理结点.若约束标志为 1,则必须将该逻辑结点分配到指定的物理结点上,否则由系统自由指派处理机.

1.3 模块说明

用程序语言(如 C、Fortran 或 Pascal)编写的能完成一定功能的顺序代码段称为模块.

```
〈模块说明〉::=〈模块〉{〈模块〉}
〈模块〉::= module 〈模块名〉
             in 〈语言名〉
             at 〈文件名〉
                〈端口表〉
〈语言名〉::= C|Fortran|Pascal
〈端口表〉::= {〈端口说明〉}
〈端口说明〉::= 〈输出端口说明〉|〈接收端口说明〉
〈输出端口说明〉::= 〈输出端口类型〉〈输出端口名〉['['缓冲区个数']'],〈输出端口名〉
                    ['['缓冲区个数']']}
〈接收端口说明〉::=〈接收端口类型〉〈接收端口名〉{,〈接收端口名〉}
〈输出端口类型〉::=outport
```

〈接收端口类型〉::=inport|pstart|astart|sstart|
〈缓冲区个数〉::=〈整数〉

outport 是输出端口,用来发送消息,仅出现在发送原语中.

inport 是输入端口,接收与它相连接的端口发来的消息,仅出现在接收原语中.

astart(alarm start port)是报警进程开始端口,仅出现在接收原语中.在接收原语执行时,报警进程处于等待消息状态,当消息一到立即剥夺执行.

pstart(periodic start port)是周期进程开始端口,仅出现在接收原语中.在接收原语执行时,标志周期进程的等待周期开始.

sstart(sporadic start port)是突发进程开始端口,仅出现在接收原语中,在接收原语执行时,标志突发事件的间隔周期开始等待消息到来激活该进程执行.

缓冲区个数定义了一个输出端口所具有的缓冲区数目,当只有一个缓冲区时此项可缺省.缓冲区越少,表示该端口的实时特性越高,缓冲区个数是用来表示实时特性的有效手段之一.

1.4 进程说明

进程是模块的实例,进程通过 create 语句创建.

```
〈进程说明〉::=〈进程〉{〈进程〉|〈子进程〉}
〈进程〉::=create〈进程名〉:〈进程类型〉:〈模块名〉〈任务说明〉
〈子进程〉::=subproc〈进程名〉:〈进程类型〉:〈模块名〉〈任务说明〉
〈进程类型〉::=period|sporadic|alarm|back
〈任务说明〉::={〈任务〉}
〈任务〉::= task〈端口名〉[pri〈优先数〉]
                    [peri〈时间〉]
                    exectime〈时间〉
                    deadtime〈时间〉
          |switch'{'
          case:〈任务说明〉
          {case:〈任务说明〉}'}'
〈优先数〉::=〈整数〉
〈时间〉::=〈整数〉
```

进程名给一个进程命名,在实时应用系统中它是一个全局量.

subproc 语句定义实时应用系统的一个子进程,它允许在同一个可执行文件中同时创建多个进程.子进程的代码段对应该 EXE 文件中的某个过程,模块名标识该过程,其特性在 module 语句中说明.

period 表示实时进程的周期进程.每次执行结束进入等待周期,一旦等待周期结束驱动进程的下一次执行.等待周期在说明 pstart 端口时定义.

sporadic 是实时进程的突发进程,在说明 sstart 端口时说明.突发进程两次执行的间隔必须大于间隔周期,它由消息驱动执行,当消息到来时,进程等待时间大于或等于间隔周期时则执行,否则进入睡眠队列.

alarm 是报警进程,由消息驱动执行.back 是后台进程,没有实时性.task 语句说明一个任务,pri 段指出任务的优先级;peri 段指出实时进程的周期时间,它只出现在周期进程或突发进程的开始任务中;exectime 段指出该任务的执行时间;deadtime 段规定了该任务的截止时间(时间死限).在 switch 语句中,只允许报警进程中的任务出现.后台进程没有实时特性,无任务说明.

1.5 端口连接

端口提供了单向信息传递机构.发送端口发送信息,接收端口接收信息,它们之间的连接方式通过 link 语句定义.

```
〈端口连接〉::= link〈输出端口〉to〈接收端口〉{,〈接收端口〉}
              |link〈输出端口〉{,〈输出端口〉}to〈突发进程开始端口〉{,〈突发进程开始端口〉}
〈输出端口〉::=〈端口〉
〈接收端口〉::=〈端口〉
〈突发进程开始端口〉::=〈端口〉
〈端口〉::=〈端口名〉|〈进程名〉.〈端口名〉
```

不同类型的端口,其连接方式不同.一个输出端口可与多个接收端口连接,一个接收端口只能与一个输出端口连接,但是,突发进程开始端口可以与多个输出端口连接.

link 语句中出现的端口必须在 module 语句中说明.module 中的端口名局部于所在模块,因此,不同的进程可以有相同的端口名,不同进程的同名端口可用〈进程名〉.〈端口名〉区分.

2 配置改变命令

在 UECnet 系统中允许用户改变系统配置,从而达到系统重构的目的.UECspec 语言提供了下述配置改变语句:

```
〈改变系统配置〉::= change〈系统名〉
                    {〈去掉连接端口〉}
                    {〈删除进程〉}
                    {〈删除模块〉}
                    {〈删除逻辑结点〉}
                    {〈逻辑结点〉}
                    {〈模块说明〉}
                    {〈进程说明〉}
                    {〈端口连接〉}
〈删除进程〉::=delete〈进程名〉
〈删除模块〉::=unload〈模块名〉
〈去掉连接端口〉::=unlink〈输出端口〉to〈接收端口〉
                  |unlink〈输出端口〉to all
                  |unlink〈突发进程开始端口〉to all
〈删除逻辑结点〉::=remove〈逻辑结点号〉
```

change 语句给出对已经存在的系统(由〈系统名〉指出)的配置改变说明.

delete 语句删除一个存在的进程,在删除该进程时必须保证该进程与其它进程之间没有连接关系.

unload 语句删除一个模块,在删除时必须保证该模块没有任何实例存在.

unlink 语句撤消端口间的连接关系.UECspec 语言提供了三种 unlink 方式.第一种是最基本的,它去掉两个端口间的连接关系;第二种是去掉与某个输出端口相连的所有连接关系;第三种是去掉与突发进程开始端口相连的所有连接关系.

remove 语句删除一个逻辑结点.执行该语句时必须保证该逻辑结点中不存在任何进程.

扩充系统通过增加逻辑结点完成.增加一个逻辑结点必须包括四部分的说明,即逻辑结

点、模块说明、进程说明及端口连接，这几部分的语法定义见 1.2、1.3、1.4、1.5.

结束语：配置说明语言 UECspec 已在分布式实时系统 UECnet 中实现. 整个系统尚属研究模型. 即使这样，我们已经发现 UECspec 对于用户描述应用用实时任务是非常重要的.

致谢　在这项研究工作中，本课题组成员给作者提出了许多建议，并给予大力支持与合作，在此致以深深的谢意！

参考文献

1　龚天富，张松梅，舒敏等. 一个分布式实时操作系统 UECnet. 计算机工程与应用，1990；(10－11)：43－47.

2　舒敏，张松梅. UECnet 配置系统. 见吴涛，张晨曦 eds.，中国青年计算机研究进展，第三届全国青年计算机会议论文集，长沙，1991，长沙：国防科技大学出版社，1991：77－80.

3　Coulas M F, MacEwen G H, Murquis G. Rnet: a hard real time distributed progamming system. IEEE Trans. on Computers, 1987; 36(8): 917－931.

4　Sloman M, Magee J, Kramer J. Builing flexible distributed computer system in Conic. In: Distributed Computiong Systems Programme, London, England: Peter Peregrinus, 1984: 86－106.

5　Kramer J, Magee J. Dynamic configuration for distributed system. IEEE Trans. Software Engineering, 1985; (11)1: 424－436.

UECspec: A CONFIGURATION SPECIFICATION LANGUAGE ——THE TOOL FOR DESCRIBING A DISTRIBUTED REAL—TIME SYSTEM UECnet

Shu Min and Gong Tianfu

(*Department of Computer Science, University of Electronic Science and Technology, Chengdu* 610054)

Abstract　Large and complicated distributed real－time systems require more flexibility, predictability and reliability besides strict timing constraints. This paper presents a configuration specification language UECspec, which is a description tool for distributed real－time application systems and is used to describe the logical network structure, communication linkage relations among processes and the real－time properties of processes at the application level of a distributed real－time system UECnet. The UECnet system performs automatically the configuration of application systems by the specifications in UECspec.

Key words　Distributed system, programming languages and software, concurrent programming, real time, configuration.