

π 演算于辅助泊车的应用^①

许超, 孙道清

(安徽师范大学数学计算机科学学院, 安徽 芜湖 241003)

摘要: 随着生活中出现停车越来越难的问题, 已出现在普适计算环境下辅助泊车的设计。在生活中有关实际问题的设计, 都需要从理论上证明。针对辅助泊车流程与兼容性的需求, 从 π 演算的语法和操作语义等理论入手, 对辅助泊车的构件行为进行建模同时验证辅助泊车设计的构件行为的兼容性。将 π 演算的理论与实际相结合, 实现了计算机理论与实践的相结合。

关键词: π 演算; 构件; 辅助泊车; 泊车; 普适计算

π -Calculus in the Application of Aided Parking

XU Chao, SUN Dao-Qing

(Anhui Normal University, School of Mathematics & Computer Science, Wuhu County, Anhui 241003)

Abstract: With more and more problems of parking in life, the design of aided parking based on ubiquitous computing has appeared. The designs of practical problems in life need to be proved theoretically. According to the demand of aided parking's process and compatibility, we use π -calculus theory such as grammar and operation semantic to model the component of aided parking's behavior and validate the compatibility of the component of aided parking. Combining π -calculus theory with practice realize a computer combination of theory and practice.

Key words: π -calculus; component; aided parking; parking; ubiquitous computing

1 引言

随着计算机的发展, 普适计算作为一种新型计算环境, 使得如今正在从“个人电脑时代”过渡到“普适计算时代”。生活中, 找到停车场停车的问题愈来愈严重, 于是, 在普适环境下辅助停车的设计也被提了出来。文献[1]提出了一种在普适环境中泊车的设计, 提出该设计的流程。在该设计中有多种构件, 构件间行为以及兼容性问题能影响整个系统等。基于此, 我们对文献[1]中构件行为使用 π 演算进行建模并验证它们之间的兼容性使文献[1]的设计可行性。 π 演算^[2]是 1991 年由图灵奖得主 Robin Milner 提出的一种以进程间的移动通信为研究的并发理论。因为其具有移动性的描述机制, 所以比较适合描述构件行为。而且, π 演算是目前比较成熟的进程代数方法之一, 它的表达能力很强。本文就

是在这样的背景下, 将 π 演算与普适计算的技术结合, 针对社会中的热点问题, 将理论应用于实践, 也实现了普适计算“以人为本”的理念。

2 π 演算概述

π 演算作为一种进程代数方法, 它在基于数学的基础上, 可以应用于建模。 π 演算能够描述流程的动态演化过程, 这是同样可以用于建模的 Petri 网模型所无法比拟的。因为其能够很好地描述结构不断变化的系统^[3], 用它来描述流程具有简洁明了的特点, 对流程的建模和验证很方便。本文就是利用了 π 演算的特点, 对辅助泊车设计的构件行为流程进行建模并验证其兼容性。

π 演算中的两个基本实体是进程和名字, 进程代表并发运行实体的单位, 每个进程有若干通道与其他

^① 收稿时间:2011-09-13;收到修改稿时间:2011-10-14

进程进行通信。^[4]当构件的行为组合是以相互交互的构件行为来描述时，通过 π 演算可以获取进程的等价和死锁的情况。因此，使用 π 演算来对构件行为建模可以避免一些未知的错误发生。

目前， π 演算由于其特点，被越来越多的人所研究。主要分两方面：① π 演算的理论研究。② π 演算作为一种建模工具在各种研究领域上的应用。前者主要是从 π 演算的理论层面，研究它的理论，诸如 π 演算的语法、验证方法等。后者主要将 π 演算的理论结合其他领域，辅助其他领域的研究，诸如人工智能、普适计算等。

π 演算的基本语法可以详见文献[5]，从文献[5]中可以看到 π 演算的数学结构很简单，这也是其能够普遍发展的重要原因。

3 辅助泊车的构件行为建模

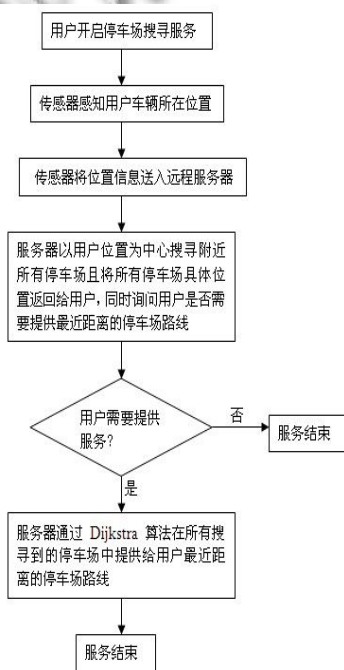


图 1 辅助泊车完整流程

文献[1]提出了一种在普适环境下辅助泊车的设计，它的完整流程如图 1。由于普适计算的应用，使得该流程中许多用户不可见的行为从生活中“消失”，可详见文献[1]中辅助泊车简化流程。该设计中，主要有两大构件：用于发送最近停车场距离的停车路线的构件和用于接收最近停车场距离的停车路线的构件。下面对这两个构件行为进行建模。将用于发送最近停

车场距离的停车路线的构件命名为 SS，它有二个输入接口：接收用户位置的接口 (User's location) 和接收提供最近距离路线的接口 (Service)。构件 SS 还有四个输出接口：附近有停车场的接口 (Bus-stops)、附近没有停车场的接口 (No bus-stop)、发送最近距离的路线接口 (The shortest route) 以及发送取消提供最近距离路线的接口 (Cancel)。如图 2 是该构件 SS 的行为视图。

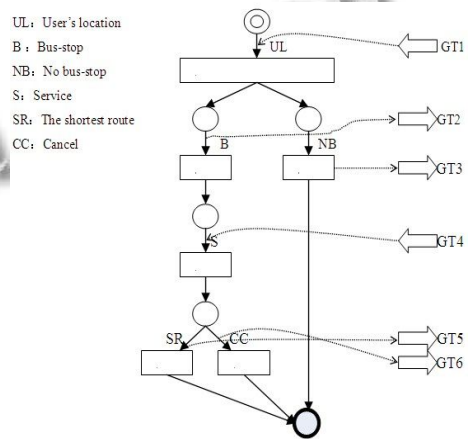


图 2 构件 SS 的行为视图

分别设置每次输入输出的通道名字，将输入 User's location (UL) 的通道命名为 GT1，将输出 Bus-stops (B) 的通道命名为 GT2，将输出 No bus-stop (NB) 的通道命名为 GT3，将输入 Service (S) 的通道命名为 GT4，将输出 The shortest route (SR) 的通道命名为 GT5，将输出 Cancel (CC) 的通道命名为 GT6。

依照 π 演算的基本语法，输入 UL 的操作可以形式化表示为 $GT1?(UL)$ ，输出 B 的操作可以形式化表达为 $GT2!(B)$ ，输出 NB 的操作可以形式化表达为 $GT3!(NB)$ ，输入 S 的操作可以形式化表达为 $GT4?(S)$ ，输出 SR 的操作可以形式化表达为 $GT5!(SR)$ ，输出 CC 的操作可以形式化表达为 $GT6!(CC)$ 。

令整个构件行为用进程 PSS 表示，附近有停车场时执行进程 PB，附近没有停车场时执行进程 PNB，提供最近距离路线时执行进程 PSR，取消提供最近距离路线时执行进程 PCANCEL。可以利用 π 演算将构件行为表示为：

$$\begin{aligned}
 PSS &= GT1?(UL).(PB+PNB) \\
 PB &= GT2!(B).GT4?(S).(PSR+PCANCEL) \\
 PNB &= GT3!(NB).0
 \end{aligned}$$

PSR=GT5!(SR).0

PCANCEL=GT6?(CC).0

针对上述的构件 SS，同样将用于接收最近停车场距离的停车路线的构件命为 RR，它有两个输出接口：发送用户位置的接口（User's location）和发送需要提供最近距离路线的接口（Service）；四个输入接口包括：接收附近有停车场的接口（Bus-stops）、接收附近没有停车场的接口（No bus-stop）、接收最近距离的路线接口（The shortest route）和接受取消提供最近距离路线的接口（Cancel）。

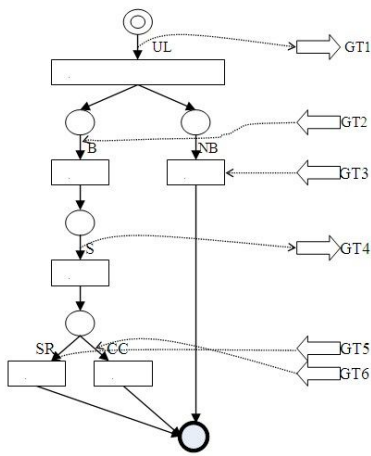


图 3 构件 RR 的行为视图

具体的构件行为视图如图 3 所示。同样可以利用 π 演算将构件行为表示为：

PRR=GT1!(UL).(PB1+PNB1)

PB1=GT2?(B).GT4!(S).(PSR1+PCANCEL1)

PNB1=GT3?(NB).0

PSR1=GT5?(SR).0

PCANCEL1=GT6!(CC).0

至此，整个构件行为的建模已经完成。可以看到，利用 π 演算不仅可以对构件行为进行建模，而且 π 演算通过其语法，使得流程的描述简洁明了。

4 构件行为兼容性分析

构件之间的动态交互相当于一个并发系统，构件行为兼容性就是主要针对构件之间的交互行为而言的，可以检查构件之间交互行为是否能正常完成操作。如果构件之间的行为是不兼容的，可能会影响到系统的稳定性和可行性。因此，对构件行为的兼容性验证

是必不可少的。同样，在文献[1]设计中，该系统的主要构件是上文所提出的构件 SS 和构件 RR，这两构件贯穿整个系统。因此，对该两构件进行兼容性分析以确保该系统的整体稳定性和可行性。下面，利用 π 演算对构件 SS 和构件 RR 之间的行为进行验证。

构件 SS 和构件 RR 之间的并行交互行为在 π 演算中的表达式如下：

$P(RR,SS)=PRR||PSS=(GT1?(UL).(PB+PNB))||GT1!(UL).(PB1+PNB1)$ ，接着进行演化：

$P(RR,SS)=PRR||PSS=(GT1?(UL).(PB+PNB))||GT1!(UL).(PB1+PNB1)$

$UL \rightarrow$
 $(GT2!(B).GT4?(S).(PSR+PCANCEL)+PNB)||GT2?(B).GT4!(S).(PSR1+PCANCEL1)+PNB1)$

$B \rightarrow$
 $(GT4?(S).(PSR+PCANCEL)+PNB)||GT4!(S).(PSR1+PCANCEL1)+PNB1)$

$S \rightarrow$
 $(GT5!(SR).0+GT6?(CC).0+GT3!(NB).0)||GT5?(SR).0+GT6?(CC).0+GT3?(NB).0)$

$SR \rightarrow$
 $(GT6?(CC).0+GT3!(NB).0)||GT6?(CC).0+GT3?(NB).0)$

$CC \rightarrow$
 $(GT3!(NB).0)||GT3?(NB).0)$

$NB \rightarrow$
 $0||0=0$

同理，另外两个输入输出序列 UL、B、S、CC 和 UL、NB 最后均变成空进程，所以三条交互路径都是可行交互路径，即构件 SS 和构件 RR 之间的交互行为是绝对兼容的。

5 MWB工具实验

MWB 工具是一个针对 π 演算开发的自动验证工具。在此，通过 MWB 工具对上述演化进行实验，证

明演算结果。如图 4-MWB 工具实验 1, 先设定各进程函数, 且 step 运行总进程函数, 得到如图 5-MWB 工具实验 2, 可以看到有 3 种可能状态。我们对所有状态进行 step 均能得出最后归为空进程。如图 6-MWB 工具实验 3 所示是其中一种结果。

```
MWB>agent Pcancel(cc)=cc.θ
MWB>agent Psr(sr)=sr.θ
MWB>agent Pnb(nb)=nb.θ
MWB>agent Ph(h,s,sr,cc)=h.s.(Psr(sr)+Pcancel(cc))
MWB>agent Pss(ul,h,s,sr,cc,nb)=ul.(Ph(h,s,sr,cc)+Pnb(nb))
MWB>agent Pcancel1(cc)=cc.θ
MWB>agent Psl(sr)=sr.θ
MWB>agent Pnl(nb)=nb.θ
MWB>agent Pbl(h,s,sr,cc)=h.s.(Psl(sr)+Pcancel1(cc))
MWB>agent Ppr(ul,h,s,sr,cc,nb)=ul.(Pbl(h,s,sr,cc)+Pnl(nb))
MWB>agent P(ul,h,s,sr,cc,nb)=Pss(ul,h,s,sr,cc,nb)|Ppr(ul,h,s,sr,cc,nb)
MWB>step P(ul,h,s,sr,cc,nb)
```

图 4 MWB 工具实验 1

```
0: !>t.<<Ph(h,s,sr,cc) + Pnb(nb)>> ! <Pbl(h,s,sr,cc) + Pnl(nb)>>
1: !>ul.<<Ph(h,s,sr,cc) + Pnb(nb)>> ! 'ul.<Pbl(h,s,sr,cc) + Pnl(nb)>>
2: !>'ul.<ul.<Ph(h,s,sr,cc) + Pnb(nb)>> ! <Pbl(h,s,sr,cc) + Pnl(nb)>>
Step>
```

图 5 MWB 工具实验 2

```
Step>0
0: !>t.<s.<Psr(sr) + Pcancel(cc)>> ! 's.<Psl(sr) + Pcancel1(cc)>>
1: !>[h=nb]>t.s.<Psr(sr) + Pcancel(cc)>>
2: !>[h=nb]>t.'s.<Psl(sr) + Pcancel1(cc)>>
3: !>t.θ
4: !>'h.<s.<Psr(sr) + Pcancel(cc)>> ! <Pbl(h,s,sr,cc) + Pnl(nb)>>
5: !>'nb.<Pbl(h,s,sr,cc) + Pnl(nb)>>
6: !>h.<<Ph(h,s,sr,cc) + Pnb(nb)>> ! 's.<Psl(sr) + Pcancel1(cc)>>
7: !>nb.<Ph(h,s,sr,cc) + Pnb(nb)>>
Step>7
0: !>'h.s.<Psr(sr) + Pcancel(cc)>>
1: !>'nb.θ
Step>0
0: !>s.<Psr(sr) + Pcancel(cc)>>
Step>0
0: !>'sr.θ
1: !>cc.θ
```

图 6 MWB 工具实验 3

(上接第 33 页)

现象, 采用 F-ALOHA 防碰撞算法, 解决传输错误概率高的问题, 缩短识别时间, 提高系统效率。设计中利用多种通信方式, 结合射频识别和单片机控制等方式, 通过仿真与调试证明了该设计方案的可行性, 因其低功耗性将在矿井等恶劣环境中逐步得到应用, 为井下安全定位系统的设计提供了很好的参考。

参考文献

- 胡洋.RFID 技术的定位算法改进及其在图书馆的应用.计算机系统应用, 2010,19(5):151-154.
- 尚亮,李文峰,李白萍.基于 MSP430F2012 和 CC1100 的主动式低功耗 RFID 标签设计.电子元器件应用, 2008,10(6):1-7.

6 小结

本文提出的 π 演算对辅助泊车构件行为的建模方法可以清楚地描述辅助泊车的流程。同时, 通过运用 π 演算的兼容性验证方法以及 MWB 工具实验的应用验证了该辅助泊车构件的兼容。通过对基于普适计算的辅助泊车设计运用 π 演算的技术, 使得辅助泊车设计中的构件避免了因不兼容而导致诸如发生未知错误的问题。将 π 演算理论运用到普适计算的具体应用中, 实现了 π 演算的高效应用。

参考文献

- 许超,孙道清.基于普适计算和 Dijkstra 算法辅助泊车的设计.软件导刊,2011(5):129.
- Milner R. Communicating and Mobile Systems: The Pi Calculus. Cambridge: UK Cambridge University Press,1999.
- Motherhood RH. Process Models. Apple Pi: Enterprise Planning and Architecture Strategies. http://www.Metagroup.com, 2009-03-15/2009-11-02.
- 吴卿,殷显煜.面向普适环境的自适应中间件模型与方法.浙江大学出版社,2010.
- Davide S, Davide W. The π -Calculus: A theory of Mobile Process.Cambridge: Cambridge University Press, 2001.

- 张长森,徐景涛,董鹏永.基于 MSP430 的矿井人员定位射频读写系统的设计.电子技术应用, 2008,34(6):60-63.
- 沈建华,杨艳琴,翟晓曙.MSP430 系列 16 位超低功耗单片机原理与应用.北京:清华大学出版社, 2004,6.
- 孙继平,李继生.煤矿井下无线通信传输信号最佳频率选择.辽宁工程技术大学学报, 2005(6):378-380.
- 任晓奎,梁朝忠.RFID 系统防碰撞算法分析与改进.计算机系统应用, 2010,19(2):93-96.
- 谢胜梅,赵军辉.基于最优化原理的 RFID 系统中的 ALOHA 防碰撞算法研究.电路与系统学报, 2009,14(4):8-12.
- 高玉芹.基于 RFID 和 WSN 的煤矿安全监控系统.仪表技术与传感器, 2007(12):76-78.