

实例一系统加油站建模

考虑一个在软件体系结构领域广泛研究的实例——自动加油站系统. 加油站中包括顾客、出纳和加油泵, 假设加油站中有充足的油料而且出纳不需要找零.

系统初始时, 加油站中只包括一个顾客、一个出纳和一个加油泵. 我们可以把这个系统抽象为顾客 1、出纳和加油泵组件, 并包括一个连接件, 其初始 SAAM 配置如图 2 所示, 组件与连接件的接口用带阴影的变迁表示, 且带有卫函数. 为了模型简洁, 我们忽略了数据类型、弧表达式和卫函数. 用 Wright 对自动加油站进行建模^[9], 需要 3 个组件、3 个连接件和 12 个附件, 建模比较繁琐. 利用 Darwin 建模时^[10], 系统中没有连接件, 顾客组件利用相同的接口付钱和加油, 容易造成冲突, 体系结构配置比较混乱, 没有正确地对体系结构进行抽象. 相比之下, SAAM 简单、直观易懂, 能保证组件的正确交互. 在 SAAM 中, 顾客、出纳和加油泵组件首先在连接件中注册并查找相应的服务组件, 连接件为它们建立 pay_1 、 pump_1 和 info 消息传递通道, 然后各个组件与相应的服务组件进行交互.

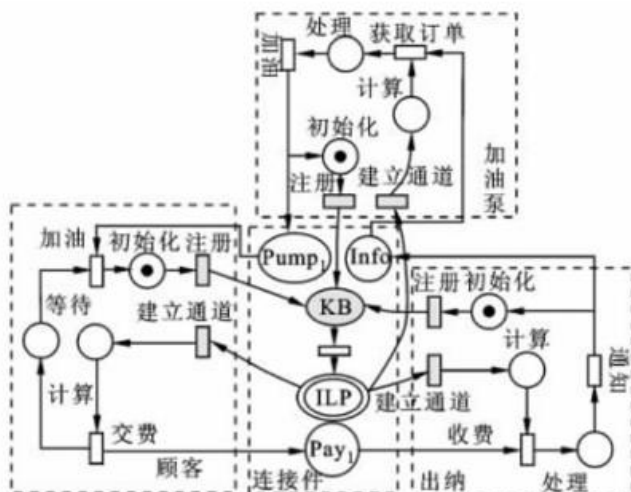


图2 自动加油站初始 SAAM

可以利用 Petri 网支持工具 INA 分析系统性质. 根据 INA 分析结果, 图 2 所示的初始 SAAM 是活的, 不存在死锁, 可以保证系统的正常运行.

在系统运行中, 加油站中不断有新的顾客到达, 系统结构呈现动态变化. 假设某时刻顾客 2 到达加油站, 则根据 2.1 节中的连接件, 利用 π 演算建立组件之间交互通道的动态过程. 顾客 2 首先在连接件中注册, 随后建立与出纳、加油泵之间的交互通道 pay_2 和 $pump_2$, 此时系统中包含 2 个顾客, 系统体系结构发生变化. 我们现在要关注的是顾客 2 到达后系统的一致性, 即系统是否能正常交互. 因此, 我们首先根据 2.2 节的体系结构一致性分析, 把新加入组件的接口和与新组件有交互行为的组件接口用进程描述出来, 然后根据接口判断新组件的加入是否会影响系统的一致性. 3 个组件的接口可以定义如下:

顾客 2 ($C_{p, cus2}$): $pay(m) = \overline{pay_2} \langle m \rangle. pay(m),$
 $inputgas(x) = pump_2(x). inputgas(x);$

出纳 ($C_{p, cas}$): $charge(y) = pay_2(y). charge(y),$

$\text{inform}(i_d, m_{sg}) = \overline{\text{info}}\langle i_d, m_{sg} \rangle. \text{inform}(i_d, m_{sg});$

加油泵($C_{p,pum}$): $\text{pump}(g) = \overline{\text{pump}_2}\langle g \rangle. \text{pump}(g),$
 $\text{getorder}(z, w) = \text{info}(z, w). \text{getinfo}(z, w).$

出纳的接口 $\text{inform}(i_d, m_{sg})$ 根据顾客的 i_d 通知加油泵给相应的顾客加油, 根据定义 2 很容易得到 $\text{pay}(m) \circ \text{charge}(y), \text{inform}(i_d, m_{sg}) \circ \text{getorder}(z, w), \text{inputgas}(x) \circ \text{pump}(g)$, 而且顾客 1 与出纳和加油泵的交互通道与顾客 2 的不相同, 不会发生冲突. 因此, 根据定理 1, 由这些接口描述的组件 $C_{p,cus2} \mid C_{p,cas} \mid C_{p,pum}$ 能正常运行, 进而根据定理 2, 由这些组件组成的系统能保持正常交互. 系统发生动态变化后, 其最终模型用 π 演算可以描述为

$$\begin{aligned} C_{p,cus1} &= \overline{\text{pay}_1}\langle m_1 \rangle. \text{pump}_1(x_1). C_{p,cus1} \\ C_{p,cus2} &= \overline{\text{pay}_2}\langle m_2 \rangle. \text{pump}_2(x_2). C_{p,cus2} \\ C_{p,cas} &= \text{pay}_1(u). \overline{\text{info}}\langle i_{d_1}, m_{sg_1} \rangle. C_{p,cas} + \\ &\quad \text{pay}_2(v). \overline{\text{info}}\langle i_{d_2}, m_{sg_2} \rangle. C_{p,cas} \\ C_{p,pum} &= \text{info}(z, w). ([z = i_{d_1}] \overline{\text{pump}_1}\langle w \rangle + \\ &\quad [z = i_{d_2}] \overline{\text{pump}_2}\langle w \rangle). C_{p,cas} \\ S_M &= C_{p,cus1} \mid C_{p,cus2} \mid C_{p,cas} \mid C_{p,pum} \end{aligned}$$

最后利用 π 演算的支持工具 MWB 分析最终的 SAAM. 分析结果表明, 最终的模型不存在死锁, 系统可以正常运行.

实例二变电综合信息管理系统。

当前,我国电力系统正在进行精简机构的改革,变电站也在朝无人、少人和一点带面的方向发展(如:一个有人值班 220KV 变电站带若干个无人值班 220KV 和 110KV 变电站),“减人增效”是必然的趋势,而要很好地达到这个目的,使用一套完善的变电综合信息管理系统(TSMIS)显得很有必要。为此,笔者组织有关力量,针对电力系统变电运行管理工作的需要,结合变电站运行工作经验,开发了一套完整的变电综合信息管理系统。

在设计 TSMIS 系统时,充分考虑到变电站分布管理的需要,采用 C/S 与 B/S 混合软件体系结构的“内外有别”模型,如图 3 所示。

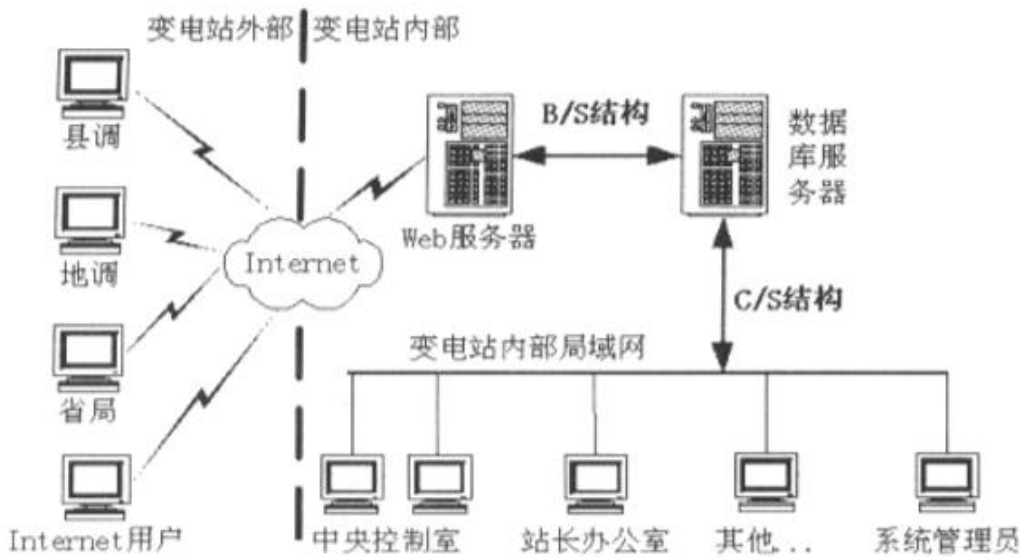


图 3 TSMIS 系统软件体系结构

在 TSMIS 系统中, 变电站内部用户通过局域网直接访问数据库服务器, 外部用户 (包括县调、地调和省局的用户及普通 Internet 用户) 通过 Internet 访问 Web 服务器, 再通过 Web 服务器访问数据库服务器。外部用户只需一台接入 Internet 的计算机, 就可以通过 Internet 查询运行生产管理情况, 无需做太大的投入和复杂的设置。这样也方便所属电业局及时了解各变电站的运行生产情况, 对各变电站的运行生产进行宏观调控。此设计能很好地满足用户的需求, 符合可持续发展的原则, 使系统有较好的开放性和易扩展性。

TSMIS 系统包括变电运行所需的运行记录、图形开票、安全生产管理、生产技术管理、行政管理、总体信息管理、技术台帐管理、班组建设、学习培训、系统维护等各个业务层次模块。实际使用时, 用户可以根据实际情况的需要选择模块进行自由组合, 以达到充分利用变电站资源和充分发挥系统作用的目的。限于篇幅, 在此, 不详细介绍各模块的功能。