

基于元胞自动机的软件架构设计

王清科¹ 马晨明² 汤晓东³

¹(上海市普陀区业余大学信息中心 上海 200062)

²(东华大学数学系 上海 201620)

³(上海市杨浦区业余大学信息中心 上海 200092)

摘要 探讨元胞自动机思想在软件架构设计领域内的应用。通过元胞抽象(Form、Controls、Operate、Process、Verify)和规则定义(显示加载规则、交互调用规则、提交卸载规则),构造出离散的可循环迭代的平行运算体系,实现普适各类业务的通用的软件架构设计。

关键词 元胞自动机 软件架构 系统设计

中图分类号 TP3 文献标识码 A DOI:10.3969/j.issn.1000-386x.2013.07.087

SOFTWARE ARCHITECTURE DESIGN BASED ON CELLULAR AUTOMATON

Wang Qingke¹ Ma Chenming² Tang Xiaodong³

¹(Information Center, Putuo Spare-Time University, Shanghai 200062, China)

²(Department of Mathematics, Donghua University, Shanghai 201620, China)

³(Information Center, Yangpu Spare-Time University, Shanghai 200092, China)

Abstract In this paper we discuss the idea of applying cellular automaton concept in the field of software architecture design. By the method of cellular abstract (Form, Controls, Operate, Process, Verify) and rule definition (display load rules, human-computer interaction rules, form unload rules), we construct a discrete and cyclic iterative parallel computing system, in this way we achieve the general software architecture design suitable for universal types of business.

Keywords Cellular automaton Soft architecture System design

0 引言

细胞通过分裂繁殖、自我复制演进出各色庞杂的生物特征,元胞自动机理论(CA)正是受此启发而建立:通过元胞抽象构造简单模型,通过演化规则的定义搭建可循环迭代的平行算法,进而演算出复杂系统^[1]。该思想现在已经成功运用于物理系统^[2],经济系统^[3],交通系统^[4],自然灾害系统^[5],历史系统^[6]等各个方面。

但是关于元胞自动机在软件架构设计方面的应用却尚不多见。对于业务类型迥异的软件系统是否也能通过元胞抽象、规则定义找到一套行之有效的通用的架构设计方法呢?经过笔者多年研究,并将其应用在公司商业软件领域,构建了公司商业软件核心架构 Foreversoft Soft Architecture(以下简称 FSA),本文将主要介绍该套体系架构。

1 元胞自动机概述

元胞自动机实质上是一种时间和空间离散的动力系统。散步在规则网格中每一元胞取有限的离散状态,遵循同样的作用规则,依据确定的局部规则作同步更新。大量元胞通过简单的

相互作用而构成动态系统的演化。元胞自动机不由严格定义的物理方程或函数确定,而采用一系列模型构造的规则构成,凡是满足这些规则的模型都可以认作是元胞自动机模型^[7]。

元胞自动机是一类模型的总称或者说是一个方法框架。其特点是时间、空间、状态都离散,每个变量只取有限多个状态,而其状态改变的规则在时间和空间上都是局部的。元胞自动机的构建没有固定的数学公式,构成方式繁杂,变种很多,行为复杂。

元胞自动机最基本的组成单位包括元胞、元胞空间、邻居及规则 4 个部分组成,其形式化定义如下^[8]:

元胞自动机是一个四元组 $\{n, S, N, f\}$ 其中:

n 为元胞空间的维数;

S 为元胞有限状态集, $S = \{s_1, s_2, \dots, s_k\}$;

N 为离散空间 Z_n 的矢量组成的 v 元组,即元胞的邻居, $N = \{x_1, x_2, \dots, x_v\}$, 其中, x_i 是邻居元胞相对给定中心元胞的位置;

f 为局部规则,其中: $f: S^v \rightarrow S$ 。

元胞自动机具有以下特征^[2,9]:

(1) 同质性 在元胞自动机模型中制定的相同演化规则决

定着其中各个元胞的演化过程。

(2) 离散性 在元胞自动机模型中,各个元胞的状态都是分离的,在时间和空间中具有离散性。

(3) 同步计算 在元胞自动机模型中,当前时刻各个元胞的状态演化都是相互独立的,而且演化过程多是同步进行的。

(4) 时空局部性 在元胞自动机模型中,当前时刻各个元胞的状态都是由其周围的邻居在前一时刻状态决定的,而邻居的确定主要是由半径所决定的,这也就是所谓的元胞自动机在时间和空间上的局部性。

近年来,为了逼真地模拟复杂的自然现象,出现了突破传统元胞自动机网格结构、实现方式限制的设计。譬如,每个元胞的状态更新函数可以是不同的,也可以在更新方式上是不同步的,甚至干脆引入概率机制灵活设计网络结构和作用方式,这样在模拟复杂现象时有更广泛的设计空间。我们称之为广义元胞自动机模型^[10]。

2 软件架构设计

现实世界中各行各业、各种领域非常之多,随着信息技术的发展和应用的不断深入,在其上的各种应用软件系统也如同自然界缤纷复杂的生物形态一般丰富庞杂。但探其究竟,我们不难发现它们也是有规可循,聚类可分,它们的内核存在着大量的一致性。

下面我们尝试将基于元胞自动机理论对其进行剖析:

2.1 元胞抽象

一般而言,实用的软件系统大多具有人机交互的功能和特性(注,诸如病毒软件这样的纯后台软件系统不在本文讨论之列)。人机交互的过程本质上是通过一系列界面的交互来实现的。它一般可表现为:界面显示、界面操作和界面提交(关闭)等三个过程。

同时,交互期间还贯穿着对操作者的授权验证,即,软件系统的用户只能看自己该看的、操作自己该操作的内容。

作为元胞分析而言,上述人机交互系统是一系列界面(Form)承载,界面内的最小单元为控件(Controls),交互操作(Operate)的实质是调用后台对应的处理过程(Process),而且交互由一系列授权验证(Verify)约束。

通过抽象,我们不难得到软件架构层面的元胞自动机对应要素:

(1) 元胞 界面(Form)中的控件(Controls)可视为自动机体系内的基本构成单元,事实上软件系统正是通过它们(lable、textbox、datagrid、button)来显示信息以及实现人机交互。

(2) 状态 控件在用户交互(Operate)之后呈现的一系列状态。

(3) 元胞空间 元胞(Controls)分布于界面(Form)之上,界面自然成为自动机体系内的元胞空间。

(4) 邻居 狭义上是指界面上与当前控件近邻的其它控件,广义上是指对影响当前控件下一时刻状态的控件。

(5) 规则 控件的状态由一系列的交互(Operate)而改变,交互的实质是处理过程(Process)的调用,处理过程便是一个状态转移函数,通过元胞当前状态及其邻居状况确定下一时刻该

元胞状态。

(6) 时间 软件体系的时间轴可以理解为一系列操作发生的顺序轴。

由于大多数应用软件都涉及数据库的应用,而数据库应用的实质一般而言是对记录的增、删、改、查询和统计等操作。这样我们就可以把软件系统划分为前后台两个域:前台界面展现交互;后台则为对数据库记录的处理。在完成了前台元胞分析之后,同样也可以对后台进行剖析和元胞分析。

对目前最为流行的关系型数据库而言,数据库单元主要体现为数据表(Table)和表内的字段(Field)。

对此,我们同样可以找到数据库层面的元胞自动机对应要素:字段(Field)可视为元胞,数据表(Table)视为元胞空间。

2.2 规则定义

如何能够让软件系统也像细胞通过自身复制演进出生物特征那样地实现呢?我们构建的公司商业软件核心架构FSA通过一套规则体系引导着软件元胞及相关的Form、Control、Operate、Process、Verify、Table和Field等要素随着人机交互过程自然地完成。

(1) 界面显示 Form加载规则:检索对应实体类,完成实例初始化;

Verify读写控制规则:交互授权验证,控制Controls可读写属性;

Controls数据提取规则:检索数据源,加载显示数据;

(2) 界面交互 Operate调用规则:检索对应业务处理过程,实现Process调用;

(3) 界面提交 Table记录处理规则:自动构造SQL语句,完成精确到Field的处理;

Form卸载规则:实例化对象卸载,缓存释放,垃圾回收。

2.3 架构实现

FSA通过六张数据表(Table)三个动态链接库文件(DLL)的复用,实现了公司所有商业项目、软件产品的框架搭建与开发。

(1) 数据表(Table)

System_Forms:实现元胞空间Form的序列化;

System_Controls:实现元胞Controls的序列化;

System_Fields:实现元胞Fields的序列化;

System_Operates:实现元胞状态触发机制Operates的序列化;

System_Process:实现元胞规则Process的序列化;

System_Verifies:实现元胞权限辅助参数Verify的序列化。

(2) 动态链接库(DLL)

FSA.Database.dll:封装了数据库处理函数;

FSA.Design.dll:封装了规则体系处理函数(加载、交互、卸载);

FSA.Business.dll:封装了通用业务处理函数(数据字典、标准控件...)

3 结语

基于元胞自动机思想构建的通用软件架构(FSA)与其它软件架构相比差异化优势体现在两大方面(共性之处不再赘述):

(1) 结构直观统一,简化系统设计 元胞抽象源自直观的界面展现,系统内部设计的映射逻辑简单清晰,不同业务领域的

应用具有高度的一致性;UI 设计一旦定型,Form、Controls、Table、Fields 等要素在设计业务的实现,而软件共性单元由元胞自动机托管。

(2) 规则明晰,提高业务专注度 FSA 通过抽象与业务解耦,实现基于表单、控件与数据模型的直接关联,开放式配置型的接口设计则可以普适业务逻辑扩展,从而尽最大可能的使程序员专注于自身差异化业务的实现,而软件共性单元由元胞自动机托管。

参 考 文 献

- [1] 赵松年. 非线性科学—它的内容、方法和意义[M]. 北京:科学出版社,1994:69-76.
- [2] Bastien C, Michel D. Cellular Automata Modeling of Physical Systems. (物理系统的元胞自动机模拟)[M]. 祝玉学,赵学龙,译. 北京:清华大学出版社,2003.
- [3] 余亮,陈荣,何宜柱. 元胞自动机与经济学应用[J]. 系统工程,2003,21(1):90-93.
- [4] Nagel K, Schreckenberg M. A cellular automaton model for freeway traffic[J]. J Phys I, 1992, 2(2): 221-229.
- [5] 宋卫国,范维澄,汪秉宏. 中国森林火灾的自组织临界性[J]. 科学通报,2001,6(1).
- [6] 韩筱璞,周涛,汪秉宏. 基于元胞自动机的国家演化模型研究[J]. 复杂系统与复杂性科学,2004,1(4):74-78.
- [7] 周恺卿,乐晓波,潘小海,等. 基于元胞自动机的线性遗传程序设计算法[J]. 计算机工程,2011,37(16):161-163.
- [8] 贾斌,高自友,李克平,等. 基于元胞自动机的交通系统建模与模拟[M]. 北京:科学出版社,2007:48-54.
- [9] 周成虎,孙站利,谢一春. 地理元胞自动机研究[M]. 北京:科学出版社,2001.
- [10] 曹兴芹. 复杂系统的元胞自动机方法研究[D]. 武汉:华中科技大学,2005.

(上接第 320 页)

密要求。

5 结 语

本文提出的 Blowfish 算法的优化方案可在 WSN 节点上正常运行。此优化方案仅对算法的实现方法进行优化,不改变算法本身结构,因此没有降低算法的安全性。并且我们是用 C 语言来实现该算法,具有很好的移植性。ZigBee 协议栈本身有安全规范,但是需要在 AES 协处理器的支持下才能完成。本文通过算法的优化,大大缩减算法执行空间,最后在 RAM 空间很小的 WSN 节点上实现了该算法,达到数据的加密要求。至于 Blowfish 算法密钥的安全性,可结合 MD5 算法实现,但所需存储空间需增大,在此不做讨论。

参 考 文 献

- [1] 钟黔川,朱清新. Blowfish 密码系统分析[J]. 计算机应用,2007,27(12):2940-2944.
- [2] 李桂满,李国. 加解密算法 Blowfish 在单片机上的应用[M]. 单片机与嵌入式系统应用,2007(10):12-14.
- [3] 尚华益,姚国祥,官全龙. 基于 Blowfish 和 MD5 的混合加密方案

[J]. 计算机应用研究,2010,27(1):231-233.

- [4] B Schneier. The Blowfish Encryption Algorithm[OL]. (2008-10-25). <http://www.schneier.com/blowsh.html>.
- [5] 刘永平. 保密传真机 Blowfish 加解密算法的实现[J]. 信息与电脑,2010(10):102.
- [6] 彭燕. 基于 ZigBee 的无线传感器网络研究[J]. 现代电子技术,2011,34(5):49-51.
- [7] 李俊斌,胡永忠. 基于 CC2530 的 ZigBee 通信网络的应用设计[J]. 电子设计工程,2011,19(16):108-111.
- [8] Sindhuja A, Logeshwari R, ThirunadanaSikamani A K. Secure PMS based on Fingerprint Authentication and Blowfish Cryptographic Algorithm[C]//2010 International Conference on Signal and Image Processing: 424-429.
- [9] 张月华,张新贺,刘鸿雁. AES 算法优化及其在 ARM 上的实现[J]. 计算机应用,2011,31(6):1539-1542.
- [10] Allam Mousa. Data Encryption Performance Based on Blowfish[C]//47th International Symposium ELMAR-2005:8-10.
- [11] Harsh Kumar Verma, Ravindra Kumar Singh. Performance Analysis of RC5, Blowfish and DES Block Cipher Algorithms[J]. International Journal of Computer Applications (0975-8887),2012,42(16).
- [12] Allam Mousa. Data Encryption Performance Based on Blowfish[C]//47th International Symposium ELMAR-2005:8-10.
- [13] 谭伟,马琪. 基于嵌入式 CPU 的 G. 722. 1 宽带语音编解码算法优化[J]. 机电工程,2010,27(11):71-74.
- [14] 王智明. PLC 基于开关量实现模拟量输出的方法[J]. 机电工程,2009,26(5):105-107.

(上接第 323 页)

中查找小字符集模式串的多模式匹配算法,即本文提出的改进算法,针对处理结果进行科学评价。

该改进算法充分利用了单模式匹配算法 MBMH 的快速移动,相比 AC 算法和 FS 算法减少了内存空间的占用,而在匹配速度和运行时间上与它们相比却有了显著的提高。最后,通过实验对其匹配速度的提高进行了分析,并通过开源网络入侵检测系统 Snort 对其性能进行了测试,证明了该算法很适合在具有大规模特征集入侵检测系统中使用。

参 考 文 献

- [1] 刘卫国,胡勇刚. DHSWM:一种改进的 WM 多模式匹配算法[J]. 中南大学学报:自然科学版,2011,42(12):3765-3771.
- [2] 王培凤,李莉. 一种改进的多模式匹配算法在 Snort 中的应用[J]. 计算机科学,2012,39(2):72-74,79.
- [3] Kanniya Raja N, Arulanandam K, Raja Rajeswari B, et al. Centralized Parallel form of Pattern Matching Algorithm in Packet Inspection by Efficient Utilization of Secondary Memory in Network Processor[J]. International Journal of Computer Applications, 2012,40(5).
- [4] Liu Zaiqiang, Lin Dongdai, Guo Fengdeng, et al. A Method for Locating Digital Evidences with Outlier Detection Using Support Vector Machine [J]. International Journal of Network Security, 2010,6(3).
- [5] 刘云峰. 模式匹配及其改进算法在入侵检测系统中的应用[J]. 电脑开发与应用,2011,24(4):41-43.
- [6] 舒银东. 基于有限状态自动机的多模式匹配算法研究[D]. 合肥工业大学,2011.
- [7] Guinde N B, Zivras S G. Efficient hardware support for pattern matching in network intrusion detection[J]. Computers Security, 2010, 29(7):756-769.