

基于免疫计算的 Multi-agent 系统设计方法

刘升, 王行愚, 游晓明

(华东理工大学信息科学与工程学院, 上海 200237)

摘要: 依据人工免疫系统的特点, 分析和设计了免疫 Agent 的结构, 建立了一种基于生物免疫机制的 Multi-Agent 系统网络模型, 并给出了其形式化描述。以此为基础, 阐述了构建具有更强的灵活性、鲁棒性和局部更新能力的复杂分布式软件系统的方法和步骤, 这对现有的软件工程起到了重要的补充作用。

关键词: 多免疫 Agent 系统; 免疫 Agent; 分布式系统; 协同进化

Design Method for Multi-agent System Based on Immune Computing

LIU Sheng, WANG Xing-yu, YOU Xiao-ming

(College of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237)

【Abstract】 According to the characteristics of artificial immune system(AIS), the structure of immune agent is analyzed and designed, the network model of multi-agent system based on biological immune mechanism and its formalization are presented. The method for constructing a complex distributed system, which will be more robust and flexible in dynamic environment and can be self-updated locally, is expounded.

【Key words】 multi-immune agent system (MIAS); immune agent (IA); distributed system; co-evolution

高度的复杂性正成为实际应用需求的特征, 这使得构造高质量的软件系统变得非常困难。这些复杂性和困难性主要体现在系统组成成分的灵活性、相互间的交互性以及对待恶意缺陷的智能防卫性等。因此, 已出现一大批软件设计方法, 比如面向对象的开发方法、基于软构件的软件设计模式和软件体系结构、基于 Agent 的分布式系统建模工具等, 尽管这些方法有着各自的优势, 但当它们面临开发复杂的分布式系统时却表现出不足: (1) 各种计算实体的交互定义得过于严格, 缺乏灵活性; (2) 缺乏足够的机制来处理复杂系统所必须具备的进化、自适应和缺陷容忍等问题。鉴于此, 软件工程必须提供一些更为有效的技术和建模方法尽可能解决这些复杂性^[1-2]及其不足。

源于生物学的计算智能技术^[2]——人工免疫系统, 在对待错误和缺陷方面开创了一种完全不同的思路, 即主动利用来自应用环境的信息, 时刻准备着各种行动去应付来自外部环境因素的挑战, 包括大范围的难以预测的事件。所以, AIS 的概念和框架已成为研究复杂软件系统设计方法的推动力, 从而帮助软件系统在遭遇恶意侵害时解决如何生存的问题。

本文设计和构建的基于免疫计算的 Multi-agent 协作系统正适应了现代计算平台和计算环境开放、异质的和分布式的特点, 可以满足现代智能计算和分布式信息处理的要求。

1 免疫 Agent 的定义及结构

免疫 Agent(IA)是一种具有免疫机理的 Agent, IA除了具有传统 Agent 的特点外, 还具有认知性、学习性和进化性等特点^[3]。

定义 1 $IA = \langle ID, APC, MEM, STA, REA, PLA, AgB, AbB \rangle$ 。

其中, ID 表示 IA 的标识, 用于唯一确定 IA; APC 表示 IA 的抗原提呈单元; MEM 表示 IA 的记忆单元; STA 表示 IA 所处的状态; REA 表示 IA 的推理决策单元; PLA 表示 IA 的规划单元; AgB 表示 IA 的标准抗原知识库; AbB 表示 IA 的抗体

知识库。其逻辑结构如图 1 所示。

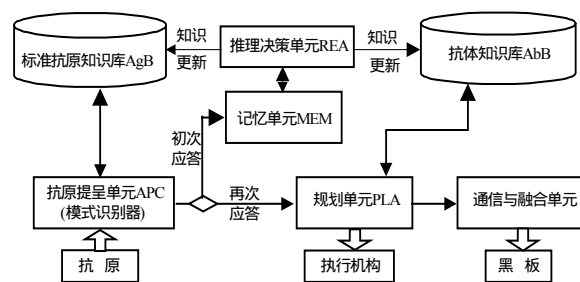


图 1 免疫 Agent 的逻辑结构

在图 1 中, 免疫 Agent 的抗原提呈单元对抗原的特征进行提取, 并和标准抗原知识库进行匹配对比, 如发现是非标准抗原, 则交给记忆单元对其进行记忆, 并交给推理决策单元。推理决策单元根据相应的进化算法, 将其产生的新的抗原特征存入标准抗原知识库, 并将产生的“抗体”存入抗体知识库中, 模拟了初次应答。若发现的是标准抗原, 则直接交给规划单元, 由规划单元提取抗体知识库中的“抗体”来驱动执行机构或通过通信与融合单元与其他 Agent 进行协调来共同抑制抗原, 模拟了再次应答。免疫 Agent 的逻辑结构体现了免疫系统应答性和记忆性的特点^[4]。

2 基于免疫计算的 Multi-Agent 系统网络模型

免疫系统是一个具有大量特殊细胞、组织和分子的分布式 Multi-Cell 系统, 每一个组成成分具有独特的物理上和功能

基金项目: 国家自然科学基金资助项目(69974014); 教育部高校博士点基金资助项目(20040251010)

作者简介: 刘升(1966-), 男, 副教授、博士研究生, 主研方向: 人工智能, Multi-agent 系统与软件工程; 王行愚, 教授、博士生导师; 游晓明, 副教授、博士研究生

收稿日期: 2006-11-30 **E-mail:** ls6601@sina.com

上的特性用来抵御易传染的疾病,免疫成分具有较高的交互性,并包括一些同等的或特殊的免疫响应方式^[6]。因此,本文构建了一种具有生物学免疫意义上的Multi-agent系统网络模型^[3-5],见图2。该免疫网络提供一系列规则用于指导和决定单个免疫Agent为适应变化的环境应采取的行为。

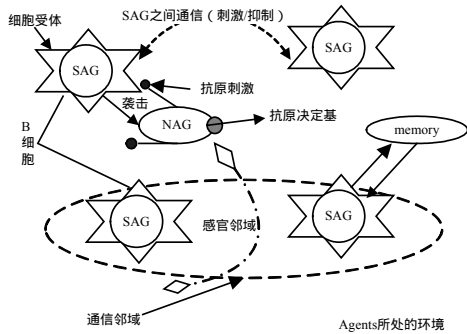


图2 基于免疫计算的 Multi-Agent 系统网络模型

根据免疫系统的特点,基于免疫计算的 Multi-Agent 系统(MIAS)可以描述为:

定义2 $MIAS = \langle ENV, Agents, F, COMM, I \rangle$ 。其中, ENV 表示 Agent 所处的环境; $Agents = SAGS \cup NAGS$ 表示细胞 Agent 集合, $SAGS$ (self agents) 表示免疫 Agent 的集合; $NAGS$ (non-self agents) 表示抗原 Agent 的集合; F 表示 Agent 之间的作用关系; $COMM$ 表示 Agent 之间的通信; I 表示 Agent 之间的相互交互。

2.1 环境(ENV)

定义3 $ENV = SAGS \cup NAGS \cup RES$ 。其中, RES 表示系统资源,这里指的是广义资源,不仅包括时间和空间,还包括一些信息(如消息等)。

2.2 作用关系(F)

Agent 之间的作用关系 $F = F^+ \cup F^-$ 。其中, F^+ 为激励或刺激作用,即 Agent 利用资源 RES 提高 Agent 之间的亲和力或增加 Agent 的数量; F^- 为抑制或袭击作用,即降低 Agent 之间的亲和力或减少 Agent 的数量。

2.3 免疫 Agent 间的通信(COMM)和基本交互(I)

源于生物学的免疫机制,系统中每一个 Agent 有其独特的行为,它们可通过通信和合作去寻找解空间,并执行一定的行为去完成不同的任务。

在图2所示的框架中, SAG 不仅在感官邻域具有识别 NAG 的能力,而且具有在通信邻域能将 NAG 的信息传递给其他 SAG 的能力,这与生物免疫网络中 B 细胞在一个特定的邻域能够辨识抗原并能将信息传递给其他 B 细胞相类似。

模型框架中 Agent 战略行为的变化可以通过它们的交互作用(Interaction: I)来进行, Agent 间的交互主要表现为在漫游区(无任何目标可检测到的区域)的浓度约束和在协同作业时的相互合作^[4]:

(1)在漫游区的 Agent 数量是受限制的,这主要是为了在 MIAS 中提高搜索效率,避免过度拥挤的 Agent 数量,浓度约束的概念源于人类免疫中自我调节的原理,该原理是利用抗体浓度的平衡机制来刺激和抑制免疫响应。假设在环境中探测的 Agent 的数量是 n ,而漫游区的浓度阈值为 r ,在浓度约束下, Agent 战略行为的变化可以表示为

$$\begin{aligned} na(E) &\rightarrow (n-r)a(D) \quad ra(E) & \text{if } n > r \\ na(E) &\rightarrow na(E) & \text{if } n < r \end{aligned}$$

其中, $na(E)$ 表示 n 个 Agent 正在探测周围环境并随机搜索目

标, $(n-r)a(D)$ 表示如果 Agent 的数量高于浓度阈值时 r , 则 $(n-r)$ 个 Agent 将被迁离所处的环境; $ra(E)$ 表示 r 个 Agent 继续探测并搜索目标。

(2)在模型框架中, Agent 也可以通过相互交互和协作,以获取共同的目标。其处理过程如下:

1)在模式识别阶段, SAGs 依靠激励函数来识别抗原的出现,通过识别函数来确定 NAGs。

2)在结合处理阶段,可以用亲和力函数来计算“自身”和“非己”行动之间的亲和力,计算方式类似于抗原决定基和抗体受体之间的亲和力计算。比如亲和力函数可以用 Euclidean 距离来定义, Agent j 和目标 i 之间在二维几何平面的 Euclidean 距离 $d_{ij}(E)$ 可表示为

$$d_{ij}(E) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

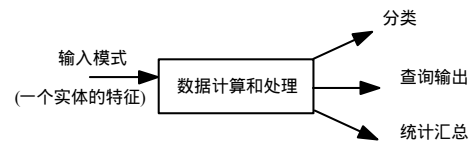
3)在活化过程中,为了模拟免疫系统,可通过设置活化阈值帮助 Agent 找到高亲和力值的行为,通常被称为成熟行为,在设计中选择高于活化阈值的亲和力值,将使 Agent 距理想目标更近。同时,也可设计结合周期作为一个指标用来衡量 SAG 去结合 NAG 所需要的时间。

4)后活化过程包括繁殖过程,此时具有成熟行为的 Agent 被繁殖,一部分被克隆的 Agent 变成记忆细胞,记忆了一些从特异的 NAG 中得到的成熟行为;其他的变成了浆细胞,在未来起保护作用。设计时可通过行为函数的设置来产生高亲和力的行为。

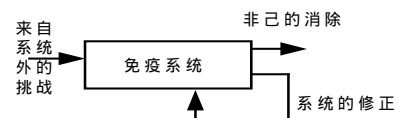
5)后处理过程包括初次和再次免疫记忆应答,一旦最终目标达到,记忆 Agent 将记住达到目标的动作执行过程。此时,即使缺少抗原刺激, Agent 间同样执行抑制,以使免疫系统达到平衡和稳定, Agent 间的抑制可通过设计具体的抑制函数来实现。

3 MIAS 的总体设计方法

在信息处理过程中, MIAS 和通常软件系统有着明显的区别,如图3所示。



(a)普通的软件系统处理系统外部数据的过程



(b)MIAS 处理系统自身的过程

图3 普通软件系统和 MIAS 的根本区别

在模式识别和数据处理方面,通常的软件系统主要用于:分类和多个类之间的映射,而 MIAS 具有如下3个主要特征^[1]:

- (1)自我维护的特征:主要涉及到不仅要监督“非己”,而且要管理系统中的“自身”及“自身”匹配者;
- (2)分布式系统的特征:Agent 间具有协同进化能力,从而形成一个具有特定识别能力的 ad hoc 网络;
- (3)自适应系统的特征:多样性与选择性,基于选择性的特征使其与通常软件系统的指令性形成对比。

同时, MIAS 只是将自身系统和与其相关的外部世界信息结合起来,并不是将所有问题的解决方法都嵌入到模型内

(下转第 251 页)