计算机科学2002Vol.29№.3

安全智体和智体群落研究*)

On Safe Agent and Agent Community

齐剑锋 彭 岩 鄢 琦 涂序彦

(北京科技大学计算机与系统科学研究所 北京100083)

Abstract Security of computer information system relys not only on security techniches, but also on the application of these techniches. This paper proposes that system security should be achieved through agent-oriented software engineering. A model for safe agent and agent community is constructed, and related concepts and properties of such a system are specifically described.

Keywords Agent, Security, Safe agent, Agent community

1 引言

计算机信息的安全日益受到关注,虽然加密、认证、访问控制等安全技术已经比较成熟,但信息安全问题仍然很严重。实际上,有了信息安全技术并不意味着解决了信息安全问题,很多安全漏洞是在软件开发时产生的[4]。必须把软件工程技术和安全技术结合起来,通过更好的软件工程过程来保证系统的安全性。

智体理论和技术为分布式开放系统的分析、设计和实现提供了新的方法。智体的概念更贴近现实世界,使用智体来分析、设计、模仿和实现现实世界中的系统,无论从概念上,还是从结构上,都更加清晰和容易。虽然不能说面向智体的软件工程适合任何环境[2],但它的确有希望成为面向对象之后的新一代软件工程方法[1]。

在本文中,我们探讨在面向智体的软件工程中保障计算机系统安全性的方法。基本思想是:系统由若干个智体构成,如果智体本身是安全的,智体之间的通信也是安全的,智体之间的关系也是安全的,则系统就是安全的;智体可以看成有生命的计算实体,通过提供合适的环境,利用智体的自主性,可以让智体自己来完成自身的安全管理。

2 智体和基于智体的系统

一般认为:智体是一种封装的计算机系统,它处于一定的环境中,具有灵活、自治的行为,从而可以完成指定的任务[5]。智体的特性可以描述如下[1]:

(1)智体是身份明确、边界清晰、接口规范的问题求解实体。

(2)位于一定的环境中,它们通过感应器接收关于状态的输入,并通过效应器对环境做出反应。

(3)为完成特定目标而设计,它们要完成的目标很明确。

(4)具有自治的特性,既可以控制内部状态,又可以控制自己的行为。

(5)在完成设计目标时,能够展现出灵活的问题求解行为。它们对环境中发生的变化能够产生及时响应;同时又能够根据时机选择新的目标。

智体的封装性和自主性对安全特别有用。智体是一个相对封闭的系统,不允许外界的侵入,智体本身应当是安全的。对于由智体构成的系统来说,保证每一个智体的安全性,是保证整个系统安全的前提。智体有自主性,它可以感知自身,可以主动地采取措施,维护自身的安全性。

如果一个计算机系统是基于智体构建的,就称之为基于智体的系统。用来描述一个基于智体的系统的内容包括[1]:(1)构成系统的智体:系统由哪些智体构成,其共同目标,各自的任务;(2)智体之间的关系:例如协作关系,主从关系;(3)智体之间的通信:智体之间需要通信从而交换协商信息。

如果保证了智体本身的安全性,并保护智体之间的关系和通信内容,就可以保证整个系统的安全性。

3 安全智体和智体群落

3.1 安全智体

如果智体的封装特性满足以下条件:1)其他智体不能够直接获得智体的内部信息;2)其他智体获得智体内部信息的唯一方法是和智体通信,向智体提出请求;3)智体拥有自己的身份;4)智体的身份可以验证,但不可以冒充;就称智体是安全的。要得到上述安全性,需要:

1)封装:采用合适的保护机制,可以是物理机制,也可以是逻辑机制,使得智体之间无法进行物理的或逻辑的直接访问。

2)通信:智体之间需要交互,交互通过语言进行,交互语言由交互协议来规范。智体之间的交互通过接口完成。理想情况下,智体只有一个接口,其思想的表达是通过语言来实现的。

3)指纹:每个智体都具有自己的身份,其身份的标志就是指纹。任何两个智体,不管它们具有多么紧密的联系,即使功能一样,其指纹都不会相同。

4)身份验证:任何两个智体之间可以相互验证身份,但验证过程并不透漏智体的身份。因此,身份验证需要采用零知识认证技术。

5)利用智体的自主性实现自我保护。

自主性是智体的重要特性,它对于智体的安全性来说非

*)本文研究得到国家自然科学基金的资助(No:60075012)。齐剑锋 讲师,博士研究生,研究方向为信息安全和软件工程。彭 岩 讲师,博士研究生,研究方向为人工智能及其应用。鄢 琦 高级工程师,博士研究生,研究方向为软件工程和人工智能。涂序彦 教授,博士导师,研究方向为人工智能及其应用。

常有价值。体现在：智体可以验证本身的完整性和可用性，在一定条件下可以实现自我修复；当智体本身被侵入或被破坏时，能够判断是否会产生对预定目标有害的行为，并采取措施，直至停止工作；智体在一定程度上可以探知环境中对自己不利的行为，并采取一定的措施；智体可以控制其他智体对自己的访问。

访问控制：访问控制和身份验证紧密相关，但是两个不同的过程。访问控制有两种：

· 预制的访问控制：这时系统初始化就是已经设计好的访问控制策略和权限。

· 经验的访问控制：智体会和其他智体交互，至于和哪些智体交互，有时候是不可预知的。智体不仅要能够和已知的智体合作，而且可以和未知的智体进行一定的合作。由于智体具有自利性，它总是根据是否有利于实现自己的目标而产生行为，因此，在和未知智体的合作过程中，智体可以评估合作的结果，是否有利于自己，从而决定是否信任该智体并继续合作。如果信任该智体，可能就会用经验知识库来修改预设的访问控制权限，向该智体提供更多的内部信息。

3.2 安全智体的本体和领地

由于安全智体的本身需要保密，因此将大大增加智体执行各种操作时的计算量。实际上，并不是智体的所有部分都需要同样强度的保护，为提高性能，把安全智体分成两部分：

· 本体：安全智体的本体部分是需要绝对保密的内容，本体的泄漏，将使智体失去安全性。例如智体需要的各种密钥、智体的基因、指纹等。

· 领地：安全智体的领地属于智体，但其他智体可以通过某种手段绕过智体读取该部分内容。但其他智体能否理解领地中的内容，取决于安全智体。如果安全智体不希望其他智体理解领地的内容，可以采取一定的保护措施。（例如，领地可能包括内存、硬盘空间、其他资源等。假设智体的领地包括若干硬盘文件，那么除了读操作，对这些文件的任何其他操作都应当由智体来完成。目前常用的操作系统尚不能提供这种机制。）

在面向智体的系统中，由于大量的操作属于智体内部的运算，因此把智体分成本体和领地两个部分，就可以由智体来对安全性和系统性能做出选择。

3.3 智体群落

多智体系统是许多个智体构成的一个集合，这些智体在逻辑上属于一个系统，它们通过协作，完成一个共同的目标，从外面看，就象一个智体。智体群落是一种多智体系统，特点是其中的智体之间强调身份识别，并可能在群落中采用某种复杂的认证机制。

如果智体群落具有如下特性：

1）本群落外的智体不能直接获得本群落的内部信息。

2）智体群落拥有自己的语言：只有本群落的智体能够听懂群落的语言。

3）每个智体都能够识别另一个智体是否属于本群落。

4）外部智体如果想得到本群落的内部信息，必须通过本群落中的一个专门智体。

那么，智体群落就是安全的。要达到上述安全性，需要：

1）封装：智体群落是由多个安全智体构成的系统，对外就像一个安全智体。前面说过，要保证智体系统是安全的，还要保证智体之间的关系和通信是安全的，即要把智体群落安全地封装起来。这种封装可以是物理的封装（外部不可得到），也可以是逻辑的封装（外部不可理解）。显然，智体群落具有和安全智体类似的模型。

2）专用语言：智体群落必须拥有自己的专用语言，只有本群落的智体可以理解，非本群落的智体不仅无法探知通信内容，即使知道了，也不能理解其含义。

3）基因：基因用来识别智体的所属和类别，因此又分成两种：群落基因、智体基因。群落基因用来识别智体是否属于本群落，不同群落的智体拥有不同的群落基因。智体基因用来识别智体是否和自己属于同一个类别，也就是，是否是从同一个智体克隆得到的。基因和指纹不同：基因用来验证智体的类别，而指纹用来验证智体的身份。由于智体群落可能是一个实时系统，也可能是一个非实时的大系统，因此指纹不可能代替基因的功能。

4）协商智体：协商智体是智体群落和外界的接口，对外来说，它是整个智体群落的代表。它必须具有足够的智能和知识，处理与外界发生的各种事件。

3.4 安全智体的模型

智体可以是软件或硬件，也可以同时包括二者。具体形式不同，所能达到的安全程度也不同。智体本质上都是逻辑，这里所做的软硬件区分，是指智体的封装形式。

软件智体实现灵活，但代码本身的保密能力有限，安全程度低。软件智体的安全性还受到所在环境安全机制的限制。软件智体可以用于构造安全程度较低的系统。由于代码分析和调试需要高超的编程能力和技巧，要花费可观的时间，因此，对很多系统来说，用安全的软件智体实现系统安全性就足够了。软件智体通常适用于非标准化设计和用量少的地方。

硬件智体的实现不仅需要良好的设计，还需要昂贵的设备和生产线来生产，其安全性取决于设计与生产过程，而不取决于使用过程，因此安全性有保障。但其结构一旦设定，就不能变动。如果有缺陷，只能重新设计新的智体，也需要重新设计生产线，代价很大。

既有软件部分，又有硬件部分的智体，结合了软件的灵活性和硬件的安全性，其适应性和安全性都较强。

无论智体的封装形式如何，都应当具有安全性需要的功能和形式，因此，具有类似的基本模型。根据前面的讨论，这个模型设计为如图1所示。为使图形简单，智体一般性的组成模块不再一一列出，都作为智体的各种功能。

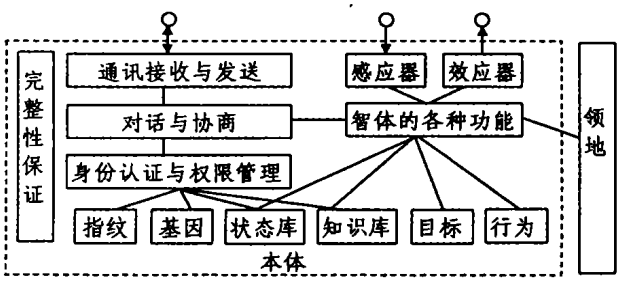

图1 安全智体的构成

3.5 智体的环境

智体存在于一定的环境中。周围的环境必须提供足够的安全特性，提供保障智体的完整性和机密性的机制。

· 保障智体的完整性：智体本身必须是完整的，不能插入其他程序，例如病毒等，否则，必然会泄漏智体的机密数据。

· 保障智体本身的机密性：智体是一个封装的实体，它要在内部进行加密、解密和认证等运算，这些运算必须是秘密进

行的，例如密钥必须保密。如果密钥泄漏了，所有的加密、解密和认证都不再有任何意义。实际上，保障智体本身的机密性不一定是保护智体的所有数据和代码，但至少要保护密钥、认证和自身检测模块。

硬件智体的机密性容易保障，而对于软件智体，必须提供能够支持认证、安全管理和访问控制等安全机制的运行环境[3]。

· 运行环境没有后门。没有任何人有特权，通过特殊的手段侵入他人的代码。

· 运行环境对智体的代码和部分数据提供保护。严格界定智体，禁止智体之间直接访问对方数据和代码。

· 允许智体对自身代码执行验证，保障自身没有被非法修改。这个功能由运行环境来验证性能更好，但允许智体验证自己，系统的可信度更大。

3.6 智体身份认证和访问控制的好处

以智体为单位执行身份认证和访问控制的好处至少有几点：

1)是对操作系统用户管理方法的补充。操作系统中的用户管理有一个基本特点，就是用户在一台计算机上注册，然后这台计算机上的所有程序都代表该用户，直到该用户注销。而在网络环境中，这个概念受到了很大挑战，在很多地方表现出不足。原因是，实际上，一台机器中的程序并不都能够代表用户，而代表用户的程序未必只在一台机器上运行。操作系统的用户管理方式和实际需要有一定的差距，对于网络应用，这种差距越来越明显。而通过面向智体的软件工程，开发基于智体的应用系统，定义出基于智体的认证和权限管理，正好适合了实际应用的需要。

2)是对操作系统文件管理方法的补充。在常用操作系统中，通常把文件操作的基本权限分为读、写两种，其中，写权限高于读权限。在实际开发企业网络中，我们经常遇到要求在允许数据共享的情况下，控制文件的复制权限。通过面向智体的软件工程，我们可以通过应用级的智体，分离读权限和复制权限，从而对复制操作进行一定程度的控制。

3)可以防止病毒危害系统。如果智体都能够执行自身完整性检测，那么病毒通过感染智体而传播的可能性将显著缩小。病毒必须作为一个智体，通过身份验证，可以控制不明身份的智体的运行，这样，就可以阻止病毒激活，并阻断其传播途径。

4)在应用系统中，支持同时处理多项业务。采用面向智体的编程和管理，把多线程作为系统处理的基本方式，每个智体都自己管理自己的所有信息。一个智体的状态变化，不会影响其它智体。用户不必学习更多的技术知识，就可以更直观、更清晰地管理自己的业务。

4 安全智体的应用

如果应用安全智体模型开发应用系统，将涉及到大量的加密、解密、认证等操作，会大大降低系统的性能，所以技术应用的主要障碍在于硬件性能。而现在，硬件得到越来越快的发展，无论是计算机的运算速度，还是网络的通信流量，都得到了极快的提高，并且还将继续提高。基于智体的应用系统的响应速度是可以接受的，而应用开发成本将显著降低。

安全智体应用的另一个条件是有适合的软硬件环境。如果是硬件智体，关键在于物理保护。如果是软件智体，就需要操作系统提供足够的保护，或者由工具程序扩展操作系统的功能，为智体构造一个能够自我保护的环境。

安全智体和智体群落思想可以应用于工业、商业、军事等很多领域的应用系统开发，这里只举3个例子。

1)用来分析和设计企业网络。企业网络可以划分为网络和网络上的应用两大部分。网络上的应用可以看成是若干个智体群落，每个应用由若干个智体构成。清晰地定义智体之间的协作、验证关系，构成安全的智体群落，即安全的应用系统。智体群落之间的关系也需要清晰的定义，构成整个企业网络，并保证其数据的安全性。这样的系统界限，功能更清晰，更接近现实世界，同时也不容易产生安全漏洞。

2)用来构造安全的遥科学系统。通常把遥科学系统分成远端系统、本地系统和通信网络，远端系统和本地系统分别由若干个子系统构成，这些子系统之间有对应关系。使用智体群落的概念，可以把某个/些本地子系统和某个/些远端子系统构成一个相对独立的系统，完整的遥科学系统由若干个这样相对独立的系统构成，这样更容易描述子系统之间的对应和所属关系，使系统边界和逻辑结构更加清晰，并避免子系统之间的相互干扰。

3)用来开发工作流系统。工作流系统用来开发流程式办公系统，这里的工作大多是流程式的，有点像流水线上产品的生产，一件工作按一定的顺序，依次经过若干个阶段的处理，最后得到一个结果。在每个处理阶段都需要验证权限。可以构造一个办公环境，包括网络和办公点，需要处理的工作就是智体。智体在某些办公点生成，它知道自己应当到哪些办公点接受处理，并根据处理结果选择去向，直到处理完毕。系统功能的设计和实现非常灵活。

结束语 本文从面向智体的软件工程的思想出发，阐述了安全智体和智体群落的概念，并利用它们来分析、设计计算机应用系统的思想。提出这种方法的目的是想在软件工程过程中，就考虑到系统的安全性问题，避免安全漏洞。目前常用的操作系统尚不能提供安全智体需要的全部机制，因此实际应用时需要自己扩展功能或采用一些折衷办法。

下一步研究工作包括分析与实现工具、安全环境的构造、系统功能的扩展等问题。

参考文献

1 Jennings N R. On agent-based software engineering. Artificial Intelligence, 2000, 117: 277～296

2 WoolDridge M J. Jennings N R. Software engineering with agents: Pitfalls and Pratfalls. IEEE Internet Computing, 1999 (May/June): 20～27

3 Roth V, Jalali-sohi M. Access control and key management for mobile agents. Comput. & Graphics, 1998, 22(4): 457～461

4 Gary M. Testing for security during development: Why we should scrap penetrate-and-patch. In: Proc. of the Annual Conference on Computer Assurance. 1997

5 刘大有，杨鲲，陈建中. Agent 研究现状与发展趋势. 软件学报，2000, 11(3): 315～321