一种信息系统生存性的量化分析框架

林雪纲[①] 许榕生[②] 熊 华[③] 朱淼良[①]
[①](浙江大学人工智能研究所 杭州 310027)
[②](中国科学院高能物理研究所计算中心 北京 100049)
[③](总参第五十四研究所 北京 100083)

摘 要 生存性是信息系统在安全性之上必需考虑的问题，对其量化分析可对系统生存性做出更为准确的评价以改进系统。基于有限状态机描述信息系统，利用系统状态转移图来定义生存性分析过程，而系统状态的层次化结构避免了Markov链模型中的列举系统状态问题。在SNA方法的基础上，提出一种便于计算机实现的生存性量化分析框架：通过系统定义、系统生存性测试和生存性计算，最后给出分析报告。其中基于事件分类分级建立的事件库使得测试方案的生成自动化和客观化，系统的生存性通过层次化的方式从可抵抗性、可识别性和可恢复性3个方面进行了量化计算。

关键词 生存性，信息系统，量化分析，分析框架
中图分类号: TP309 **文献标识码**: A **文章编号**：1009-5896(2006)09-1721-06

A Framework of Quantitative Analysis for Information System Survivability

Lin Xue-gang[①] Xu Rong-sheng[②] Xiong Hua[③] Zhu Miao-liang[①]
[①](*Institute of Artificial Intelligence, Zhejiang University, Hangzhou* 310027, *China*)
[②](*Computing Center, Institute of High Energy Physics, CAS, Beijing* 100049, *China*)
[③](*The Fifty-fourth Institute, Headquarters of the General Staff, Beijing* 100083, *China*)

Abstract Survivability should be considered beyond security for information system, and quantitative analysis can assess system survivability accurately for improvement. Information system is presented by finite state machine and its state transition map is used to describe analysis process, where the hierarchical structure of system state avoids the problem of enumerating states in Markov chain model. Based on SNA method, a framework of quantitative analysis is introduced: defining system, testing system's survivability, computing survivability, and giving analysis report finally, which is easily implemented by computer. In the framework, the event database which is based on event classification and grade makes creating test project automatically and objectively, and survivability is computed through resistance, recognition and recovery in a hierarchical process.

Key words Survivability, Information system, Quantitative analysis, Analysis framework

1 引言

随着计算机和网络的快速发展和广泛应用，关系到社会正常运作的大部分基础设施是由计算机和网络来组成和控制的，而信息系统(Information System, IS)又在其中扮演着十分重要的角色，因此信息系统的安全和可靠性直接影响到电力系统、金融系统等关键基础设施的运行。生存性(Survivability)是信息系统在安全性的基础上必须面对的问题，因为其关注的是系统在遭受到恶意攻击、软硬件故障以及其他突发事故时，还能否继续提供满足用户需求的服务。系统不可能获得百分百的安全，生存性就是考虑系统的安全性被破坏后的性能情况，而生存性分析可以对系统服务在遭受这些事件后的运行情况进行评价和判定，并给出相应的改进建议，以便能在有限的资源下尽量提高其生存性。

生存性概念虽然在1990年就已经提出，但更多的是针对通信网络硬件和链路进行研究，而对于信息系统的研究较少。国外主要是Carnegie Mellon大学的SEI，CERT/CC的ISW以及Virginia大学等；而国内的研究结构主要是航天科工集团二院、国防科技大学等单位。信息系统的生存性分析过程是一个复杂的工程。可生存性网络分析(SNA)[1]方法给出了分析的一般过程步骤，甚至定义了分析小组和客户小组的工作流程，最后将分析结果报告提交给用户。该方法虽然提供了一个分析框架并进行了一些实际工程应用[2]，但并没有对生存性进行量化分析；而且分析过程都是人工操作，分析结果受人为因素的影响较大。此外，部分学者在理论上提出了信息系统的生存性分析方法，如Jha[3]将系统网络结构中的节

2004-12-31 收到，2005-12-30 改回
国家 “973”项目（G1999035806）资助课题

点都用一个有限状态机来表示，综合利用模型检查、贝叶斯分析以及概率系统等数学方法进行定性和定量分析。Gao[4]提出对于每一个系统服务，它是由一些不同影响权重的关键规格来描述其生存性的；而对于每个关键规格，可以通过不同方法来实现，每个方法都有优先级。每个方法又是由若干组件提供的服务组成的，这样整个系统的生存性就可以通过各个组件的生存性来表示。Linger等[5]提出一个FSQ框架来表示复杂系统，通过利用流、服务和质量属性 3 个方面的描述来评估和管理系统生存性。郭渊博[6]利用分布式系统中服务和配置与配置和配置之间的支持和依赖关系，通过对配置的定量描述来定量刻画服务的可生存性。

上述这些研究工作或工程上进行定性分析，或理论上进行了定量分析而没实际应用，都没有涉及到如何利用计算机进行辅助的量化分析。而生存性分析类似于计算机信息系统安全等级保护，系统生存性最终也将实现等级划分和认证，因此有必要尽量减少分析过程中的人为因素。在本课题组的生存性分析模型[7]的基础上，本文利用有限状态机对信息系统进行描述，并利用状态转化图来对表述生存性分析方式，通过系统状态的层次化表示，提出了一种具有很好可行性的生存性量化分析框架。具体内容组织如下：第 2 节介绍了一种基于有限状态机的分析模型；第 3 节描述了生存性分析框架，其中重点介绍了量化分析方式。

2 分析模型

利用系统科学的概念，信息系统可以看作是一个开放的复杂系统，而生存性作为其一个固有属性不仅受到系统自身状况的影响，而且也与其运行环境有关。

可用如下的有限状态机(FSM)来对信息系统进行描述：$M=(K,\Sigma,\delta,q_0,F)$，其中，$K=\{q_0,q_1,\cdots,q_{N-1},\}$ 为包含 N 个系统状态的状态集，而系统状态可根据系统提供的基本服务(Essential Service, ES：系统遭受攻击、故障等情况下还必须能提供的服务)的运行状况来划分；$\Sigma=\{\mathrm{FS}_0,\mathrm{FS}_1,\cdots,\mathrm{FS}_{M-1}\}$ 为包含 M 个故障情景的集合，所谓故障情景(Fault Scenario, FS)是为达到影响系统服务的某个意图(intention)而发生的一系列事件；q_0 为系统初始状态，$q_0\in K$；F 为系统的最终稳定状态集，系统的稳定态可能是 K 中的任何一个状态，故 $F\subseteq K$。对系统输入故障情景Σ，由状态转化函数 $\delta: K\times\Sigma\to K$ 可得到系统下一个状态。对于大部分的系统来说，通常可通过对系统输入一些激励，观察系统的响应来进行性能测试和分析，因而故障情景可视为对系统的激励，而系统状态的变化为响应。

根据Ellison的生存性定义，信息系统的生存性可以通过其可抵抗性、可识别性和可恢复性(简称 3R：Resistance, Recognition, Recovery) 3 个方面来描述，可抵抗性反映了系统生存性的基本需求，即系统的安全性，抗攻击能力；可识别性反映了系统对自身状态和环境的监视能力；可恢复性则描述了系统的自我修复能力和自适应能力。这 3 种性能从另一方面也反映了现代可生存性系统的设计思想：防护、监视和反应[8]，因此本文也从这 3 个方面来定义生存性。通过利用FSM对信息系统的描述，其状态变化情况为图 1(a)所示，假设系统有从高到低 $q_i(i\in[0,4])$ 5 个状态，其中 q_0 为系统正常工作状态，q_4 为系统瘫痪状态，则图中实线表示为系统在如图中 FS_i 所示故障情景作用下的状态变迁，即系统的可抵抗性；图中虚线为系统在故障情景作用后因为系统的重配置等措施使得状态有所恢复，即系统的可恢复性；而系统对图中实线变迁的识别检测能力则表示了系统的可识别性。

假如直接利用图 1(a)来对系统的生存性进行分析，不可避免地将涉及到如何对系统的状态进行定义并划分穷举出来，而这利用计算机来实现几乎是不可行的，而且难以实现标准化和通用化。这个问题存在于利用Markov模型[3]的生存性分析中，因此必须避免进行实际的系统状态切割。根据定义，系统状态是由系统基本服务的状态来决定的，而基本服务又受故障情景的影响。对于每个故障情景，其意图一般可划分为若干个子意图，即分为几个步骤由一组事件来完成，其中仅由一个具有具体目标(Target)的事件完成的步骤称为原子任务(Atomic Mission, AM)，故障情景最终都可以由一些原子任务而组成。对于每个原子任务来说，可以选择不同的事件(Event)来完成其目标，因而系统在图 1(a)中遭受 FS_i 的作用最后可用这些事件来表示。故系统状态可用如图 1(b)所示的 4 层结构依实线箭头逐层来表示，而对系统状态的分析可从底往上逐层进行，这样就避免了图模型[9]和Markov链模型[3]中需要定义系统的状态集来分析生存性，而转化为从事件、原子任务、故障情景和基本服务来分析系统的生存性。

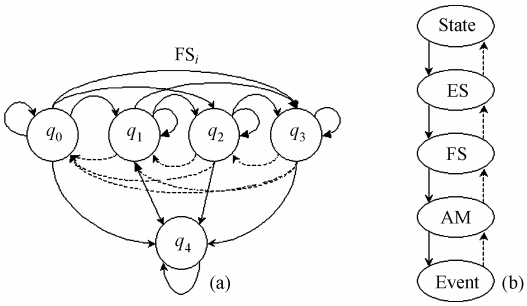

图 1 (a) 系统状态转移图 (b) 系统状态层次结构

Fig.1 (a) System state transition
(b) Hierarchical structure of system state

3 量化分析框架

信息系统的生存性分析是一个复杂繁琐的工程，为此，基于上述的分析模型，在 SNA 方法的基础上，针对实现计算机辅助分析和最终量化评分的需求提出了一种改进的分析框架，具体分析流程如图 2 所示，整个分析框架主要包括 7 个步骤。

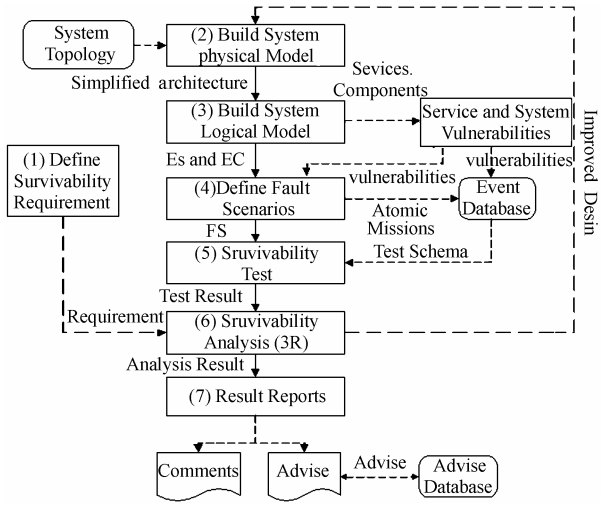

图 2　生存性量化分析框架

Fig.2 Framework of quantitative survivability analysis

3.1　生存性需求定义

对于信息系统来说，并不是都要求具有很高的生存性，而应该根据具体服务需求、系统运行环境和资金预算来决定系统需要怎么样的生存性。因此，事先定义系统需要怎样的生存性，即系统的生存性级别，以此为基准对比实际的生存性分析结果，从而判定系统是否达到其生存性要求。

此外，系统的生存性是相对于其运行环境而言的，同样的系统在不同的攻击者面前呈现出不同的生存性性能。系统的生存性需求也是随着系统运行环境的改变而不同，如同一个系统在战争时期与和平时期的生存性要求不同。故系统生存性需求可以形式化定义为 REQ={EV,Sur'}，其中 EV 为系统运行环境，具体定义在生存性测试中描述，Sur′=[Resis′,Recog′,Recov′] 为系统的生存性需求值，采用向量形式描述，其分量为系统生存性指标，依次为可抵抗性、可识别性和可恢复性值。

3.2　系统物理模型的建立

信息技术的快速发展使得系统组件和系统服务更加多样性，网络的广泛应用使得系统规模不断增加甚至无界，系统用户变得不确定。系统生存性关注的是整个系统提供服务的能力，而非单个组件的性能，因此没必要也不可能对系统的每个细节进行分析。

根据系统设计说明书和系统拓扑图，以系统服务和系统用户需求为主线对系统进行简化，忽略一些与系统服务联系不大的物理组件，从而建立一个比较简洁的系统物理模型。

3.3　系统逻辑模型的确立

上步简化了系统的物理结构，但还需对系统的内部逻辑结构进行定义，以更好地分析系统生存性与系统服务的关系。逻辑组件是指提供某项服务的逻辑单元，系统物理模型中的一个物理组件可能包含多个逻辑组件。

根据用户需求定义系统的基本服务，所有基本服务组成一个基本服务集 $\{ES_i\}$。利用工作流的概念，可以将基本服务从最终的服务端到用户端的路径视为一个管道，其中运载着的就是基本服务流。将基本服务流 ES_i 经过的系统逻辑组件标识出来，根据定义这些逻辑组件组成了该服务所需的基本组件集 $\{EC_{ij}\}$，即这些基本组件组成了该“工作流”中的参与方(Participant)。根据系统运行所处环境和基本服务运行状况，选择针对 ES_i 的若干个典型故障情景组成一个故障情景集 $\{FS_{ij}\}$，每个故障情景根据其对基本服务的危险度都有一个危害性权值 P_{ij}，同样每个故障情景 FS_{ij} 形成的故障流对应了另一个组件集 $\{EC'_{ij}\}$，通常存在 $\{EC'_{ij}\}\subseteq\{EC_{ij}\}$。因此，信息系统的基本服务、故障情景和基本组件之间形成了如图 3 所示的层次关系，即系统的逻辑模型。

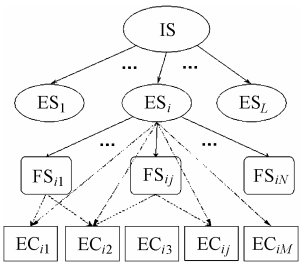

图 3　ES-FS-EC 关系图

Fig.3 Relation of ES-FS-EC

系统基本服务一般可划分为若干个子服务，再依次划分这些子服务直到其可由某单一基本组件来完成，这种子服务称为原子服务(Atomic Service, AS)。故基本服务可定义为如下形式：$ES_i=\{dss_i,vs_i,AS,EC,FS\}$，其中 dss_i 为该基本服务的描述，vs_i 为表示该服务的重要性权值，AS 为组成它的原子服务集 $\{AS_{ij}\}$，EC 为支撑它的基本组件集 $\{EC_{ij}\}$，FS 为影响该基本服务的故障情景集 $\{FS_{ij}\}$，具体形式将在下节定义。而每个原子服务也可类似定义为 $AS_{ij}=\{dsa_{ij},va_{ij},AM\}$，其中 va_{ij} 为该原子服务在 ES_i 中的权重，$AM=\{AM_{ijk}\}$ 为影响此原子服务的故障情景中的原子任务组成的一个集合。对于基本组件，可形式化定义为 $EC_{ij}=\{dse_{ij}, IF,ES,FS\}$，其中 ds 为该基本服务的描述，$IF=\{IF_{ij}\}$ 为组件操作系统、软件版本和漏洞等信息(在生成测试方案时将利用到)，ES 为其支撑的基本服务集 $\{ES'_i\}$，FS 为涉及到该基本组件的故障情景集 $\{FS'_j\}$。

3.4　故障情景定义

生存性区别于可靠性的一个重要特点就是其分析的事件是相关的，而可靠性一般假设各种事件之间是独立的。在信息系统的生存性研究中，人们更关注于人为的恶意事件对系统生存性的影响，为了达到影响系统某个基本服务的目的，入侵者总是采取一系列的步骤，发起一系列的事件来对系统进行探测、攻击、设置后门等。为此，这里将这些相关的、具有共同意图的事件用故障情景来表示。

根据系统基本服务的运作流程、环境、日常入侵及故障情况来分析该系统基本服务的缺陷和漏洞，从而对每个基本服务选择若干个典型的故障情景并对其关键步骤进行定义，组成该基本服务的故障情景集。

综合上述步骤的描述，故障情景可以定义为 $\mathrm{FS}_{ij}=\{\mathrm{dsf}_{ij}, \mathrm{ES}_i, P_{ij}, \mathrm{AM}, \mathrm{EC}'\}$，其中 dsf_{ij} 为该故障情景的描述，ES_i 为被影响的基本服务，P_{ij} 为危害性权值，AM 为故障情景的原子任务集 $\{\mathrm{AM}_{ijk}\}$，即定义了 FS_{ij} 的若干个关键步骤，EC′ 为该故障情景涉及到的基本组件集 $\{\mathrm{EC}'_{ij}\}$。对于每个原子任务来说，也可以定义为 $\mathrm{AM}_{ijk}=\{\mathrm{dsm}_{ijk}, I_{ijk}, \mathrm{AS}, E\}$，其中 dsm_{ijk} 为该原子任务的描述，I_{ijk} 为表示该原子任务在 FS_{ij} 中的权重，$\mathrm{AS}=\{\mathrm{AS}_{ijkl}\}$ 为该原子任务作用的原子服务集，E 为可以完成该原子任务的各种事件集 $\{E_q\}$。AM 中元素之间关系对于故障情景是逻辑“与”，而 $\{E_q\}$ 中元素关系对于原子任务来说是逻辑“或”。

从基本服务、基本组件、故障情景的定义可以看出，这 3 个集合中两两之间元素都是多对多的关系，而原子任务与原子服务之间也是一个多对多的关系，这样就形成一个链表数据结构，便于计算机程序的实现。

3.5 生存性测试

在故障情景的定义中，仅仅定义了其意图的实现方案，即其原子任务集 $\{\mathrm{AM}_{ijk}\}$，并没具体给出每个原子任务的具体实现方法，即事件集 $\{E_q\}$。系统在运行时会遭遇各种不同的事件，这些事件随着环境的变化也改变。因此，可将事件分为不同的级别，不同级别的事件组合可以表示不同的运行环境，即各个原子任务可由不同事件来完成。

为了进行事件的分级，需要提出一种针对事件属性的量化指标，为此定义事件的 7 个分级属性为：发起该事件工具的可用性、事件发起难度、必需资源、对应漏洞情况、事件附带后果、事件的恢复代价和发生该事件的前提条件，并且定义各个属性值为 $a_i(\in[0,1])$ 以及各属性对应的重要性权值 $v_i\left(\sum v_i=1\right)$，则该事件的危害性值可定义为 $w=\sum a_i\cdot v_i$。定义若干个临界值 $L=\{L_q\}$，对比事件危害性值和 $\{L_q\}$ 可将事件分为若干个级别。此外，定义每个级别事件的分布权值为 $T=\{t_q\}$，表示原子任务选择该级别事件来完成的概率。由此可见，调节级别分布权值 $\{t_q\}$ 和级别临界值 $\{L_q\}$，则表示利用不同的事件集 $\{E_q\}$ 来完成原子任务，即不同的环境。因此，需求定义中的环境可以定义如下 $\mathrm{EV}=\{L,T\}$。

在实现上述分级的同时还基于一种面向对象(Target)和意图(Intention)层次化的事件分类方法(具体分类方法见文献[10])进行分类，从而将常见的事件组织成一个事件库。根据故障情景定义中的原子任务定义以及组件缺陷信息 $\{\mathrm{IF}_{ij}\}$ 等搜索事件库来选择各种攻击等事件，从而组成故障情景中原子任务的事件集。这样，就形成了系统的生存性测试方案，类似于系统状态的层次结构，该方案也具有“事件-原子任务-故障情景-基本服务”的层次结构。

根据生成的测试方案对系统每个基本服务依据其故障情景集进行测试，这种测试方法可以对真实系统进行实际的测试或开发一种计算机模拟分析平台[11]对系统的现有策略和事件发生概率进行模拟和测试，测试结果作为量化分析中需要的参数，而通过事件库的建设使得测试方案中事件集的生成自动化和客观化。

3.6 生存性分析

通过对系统的生存性测试，可根据一定的计算模型对结果进行整合，从而得到系统的生存性值。基于分析模型中系统状态的分层结构以及生存性测试中的层次结构，可以从下面 3 个方面对系统的生存性进行层次化的量化分析。

3.6.1 可抵抗性(Resistance) 可抵抗性反映了系统提供的基本服务对各种事件的抵抗能力，强调的是系统提供的基本服务而不是单个组件的性能。根据故障情景的定义，对于系统基本服务 ES_i 的故障情景 FS_{ij}，其危害性权值为 P_{ij}，I_{ijk} 为其原子任务 AM_{ijk} 的权值。为完成原子任务，从级别 q 中选择 N 个事件组成该级别的事件集 $\{E_{qn}\}$，所有级别的事件组成原子任务的事件集 $\{E_q\}$。而 $\{E_{qn}\}$ 的危害性值可定义为 $W_q=\left(\sum_{n=1}^{N} w_{qn}\right)\Big/N$，测试可得到该级别事件的成功率为 $\mathrm{Pr\_Fault}_q$。

故原子任务 AM_{ijk} 的可抵抗性可定义为：$\mathrm{Resis\_AM}_{ijk}=\sum_q\left(W_q\cdot\left(1-\mathrm{Pr\_Fault}_q\right)\cdot t_q\right)$。根据各原子任务之间的逻辑“与”关系和权值，可得到故障情景 FS_{ij} 的可抵抗性为 $\mathrm{Resis\_FS}_{ij}=\sum_k(\mathrm{Resis\_AM}_{ijk}\cdot I_{ijk})$。综合所有的故障情景可得到基本服务 ES_i 的可抵抗性为 $\mathrm{Resis\_ES}_i=\sum_j(\mathrm{Resis\_FS}_{ij}\cdot P_{ij})$。

根据基本服务的定义，每个服务的重要性由一个关键性权值 vs_i 表示，因此整个系统的可抵抗性可表示为 $\mathrm{Resis}=\sum_i(\mathrm{Resis\_ES}_i\cdot\mathrm{vs}_i)$。

3.6.2 可识别性(Recognition) 生存性分析范畴内的可识别性不同于入侵检测系统(IDS)，后者更多地关注单个事件的识别，而前者强调对整个系统状态的识别。为识别出图 1(a)中实线所示的状态转移，可以根据系统状态的层次结构，依次转化为对基本服务性能下降的识别、对故障情景的识别和对各个事件的识别。对于单一事件，系统越早识别越好，而且越在边界和底层上识别越好，比如针对服务器的攻击在防火墙识别出来比通过服务器主机日志识别好，因为此时攻击已经发生在该主机上了。对于每个故障情景来说，越在前面的原子任务识别出来越好，权重大的原子任务越该被识别。而对于基本服务，则较重要的服务应该性能监控的更严密些。

根据测试结果，原子任务事件集中 q 级别事件的识别率 $\mathrm{Pr\_Rg}_q$ 可简单定义为被系统识别出来的事件占该级别所有事件的百分比，而这种识别可通过系统中的 IDS、系统日志或系统管理员的人工管理来完成，因此原子任务 AM_{ijk} 的可识别性可定义为 $\mathrm{Recog\_AM}_{ijk}=\sum_q(W_q\cdot\mathrm{Pr\_Rg}_q\cdot t_q)$。根据各原子任务之间的权重，可知故障情景 FS_{ij} 的可识别性为 $\mathrm{Recog\_FS}_{ij}=\sum_k(\mathrm{Recog\_AM}_{ijk}\cdot I_{ijk})$。

综合基本服务 ES_i 的所有故障情景，可得 ES_i 的可识别

性为 $\mathrm{Recog\_ES}_i = \sum_j (\mathrm{Recog\_FS}_{ij} \cdot P_{ij})$。因此，所有基本服务的可识别性组成了整个系统的可识别性： $\mathrm{Recog} = \sum_i (\mathrm{Recog\_ES}_i \cdot \mathrm{vs}_i)$。

3.6.3 可恢复性(Recovery)　虽然系统在故障情景的作用下向低状态转变，但这种转变可能是暂时的，由于系统的重新配置或冗余等措施，系统可能恢复原状态或转移到另一个中间状态，系统的可恢复性反映了系统基本服务或者全部服务受影响后能否恢复以及能恢复到什么程度。

各个服务因为服务等级和对象的不同，其恢复时间需求也不同，如实时性高的服务需要最短时间内恢复。定义基本服务 ES_i 中原子服务 AS_{ik} 的恢复时间需求为 $\mathrm{Req}T_{ik}$，在 FS_{ij} 的原子任务 AM_{ijl} 作用下实际测试时间为 $\mathrm{Test}T_{ijl}$。在测试时间内，系统服务的恢复程度为 $\mathrm{Pr\_Rv}_{ijl}$，该值可定义为该服务能实时完成的请求占用户所有请求的百分比，则折算成的实际恢复程度为 $(\mathrm{Pr\_Rv}_{ijl} \cdot \mathrm{Req}T_{ijl} / \mathrm{Test}T_{ijl})$。

故原子服务 AS_{ik} 的可恢复性可表示为需求恢复时间、实际恢复时间、恢复程度以及对应的原子任务权值的函数：$\mathrm{Recov\_AS}_{ik} = \sum_j \sum_l (I_{ijl} \cdot (\mathrm{Pr\_Rv}_{ijl} \cdot \mathrm{Req}T_{ijl} / \mathrm{Test}T_{ijl}))$。

基本服务 ES_i 的可恢复性则为 $\mathrm{Recov\_ES}_i = \sum_k (\mathrm{Recov\_AS}_{ik} \cdot \mathrm{va}_{ik})$，综合所有基本服务的可恢复性得到系统可恢复性为 $\mathrm{Recov} = \sum_i (\mathrm{Recov\_ES}_i \cdot \mathrm{vs}_i)$。

上述计算涉及到的参数或事先根据专家经验定义，或由生存性测试得到，而整个 3R 量化分析过程，可组织成如图 4 所示的层次结构，其中箭头上的标识表示各层计算的主要影响因子，这种自下而上逐层权值式的分析计算方式条理清晰，便于数据结构的定义，适合计算机程序实现。

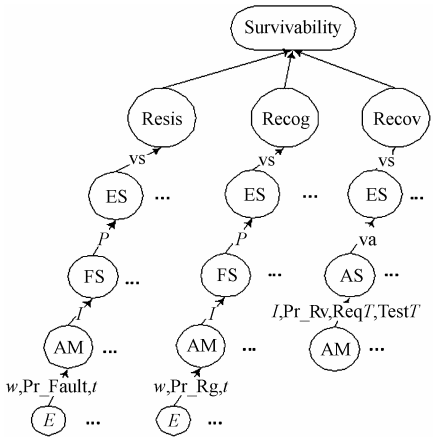

图 4　生存性的量化分析过程

Fig.4 Quantitative analysis process

3.7 分析结果报告

根据上步的计算结果 $\mathrm{Sur} = [\mathrm{Resis}, \mathrm{Recog}, \mathrm{Recov}]$，对比系统的生存性需求 $\mathrm{Sur}' = [\mathrm{Resis}', \mathrm{Recog}', \mathrm{Recov}']$ 可判定系统设计是否达到了要求。类似于事件库，系统生存性的常见策略也可组织成一个建议库。根据生存性测试的各个条目结果，对每个基本服务、每个故障情景的情况，搜索建议库提出相应的改进建议。这样，将整个生存性分析过程以及分析结果和改进建议组织成一个报告文件，形成最后的分析报告提供给用户。

当然，进行系统生存性分析的目的是改进系统，因此可根据分析结果对系统设计进行改进，再对新的设计进行生存性分析，依次循环测试量化系统生存性来验证改进情况，而且判定不同改进之间的性价比(提高的生存性值/投资额)，从而决定采取哪种改进方式。

4　结束语

继系统安全之后，信息系统的生存性已成为另一个关注点。然而，信息系统的生存性研究整体上还不是很完善，也没形成如同安全评估的标准，尤其是在生存性定义和量化分析方面[12]急需给出明确的规范。本文在信息系统的生存性分析方面做出了尝试，提出一个量化分析框架，定义了针对信息系统生存性分析的流程规范，该框架实质上是通过系统定义(物理模型及逻辑模型定义)、环境定义(故障情景定义)、系统测试(生存性测试)及结果分析(生存性分析及分析结果报告)来完成的。对系统状态的层次化表示避免了状态的直接定义与分析，而基于事件的分类分级建立的事件库便于测试方案的生存，这样就使得该分析框架具有较好的计算机分析自动化前景，这是本文分析模型和分析框架区别于以往研究成果的地方。基于该分析框架，本课题组进行了实际系统的演示分析[13]，并进行生存性分析平台原型系统的开发，以便利用计算机来减少分析工作量以及实现分析的客观化。当然，要应用于实际的工程分析，该分析框架还需要进一步的细化，尤其是计算过程中参数的进一步细化以及事件库中大量事件的整理。

参 考 文 献

[1] Ellison R, Fisher D, *et al*.. Survivable network systems: an emerging discipline. Technical Report CMU/SEI-97-153, 1997.11

[2] Ellison R J, Linger R C, *et al*.. A case study in survivable network system analysis. Technical Report CMU/SEI-98-TR-014, 1998.9

[3] Jha S, Wing J, Linger R, Longstaff T. Survivability analysis of network specifications. Proceeding of International Conference on Dependable Systems and Networks, New York, 2000.6: 613−622.

[4] Gao Zhixing, Ong Chen Hui, Tan Woon Kiong. Survivability assessment: modeling dependencies in information systems. 4th Information Survivability Workshop (2001/2002).

[5] Linger R, Hevner A, *et al*.. Semantic foundations for survivable system analysis and design. Proceedings of the International Conference on Dependable Systems and Networks, Goteberg, Sweden, 2001.7.

[6] 郭渊博, 马建峰. 分布式系统中服务可生存性的定量分析. 同济大学学报, 2002, 30(10): 1190−1193.

[7] Lin Xuegang, Xu Rongsheng, Zhu Miaoliang. Survivability analysis for information systems. Proceedings of the 7th International Conference on Advanced Communication Technology, Phoenix Park, South Korea, 2005.2, 1: 255−260.

[8] Harrison W S, Krings A W, *et al*.. On the performance of a survivability architecture for networked computing systems. Proceedings of 35th Hawaii International Conference on System Sciences, Big Island, Hawaii, 2002.1: 1−9.

[9] Krings A W, Azadmanesh M H. A graph based model for survivability analysis. Technical Report UI-CS-TR-02-024, 2004.

[10] 林雪纲, 许榕生. 计算机和网络事件的分类分析及应用. 2005 年中国计算机网络安全应急年会论文集, 广西桂林, 2005.3.

[11] HyungJong Kim. System specification based network modeling for survivability testing simulation. Proceeding of 5th International Conference on Information Security and Cryptology, Seoul , Korea, 2002: 90−106.

[12] Westmark V R. A definition for information system survivability. Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), Big Island, Hawaii, 2004.1: 303−312.

[13] 孙巍, 林雪纲, 钱桂琼, 许榕生. 网上选课系统的生存性分析. 中国信息协会信息安全专业委员会2004年年会论文集,湖南, 张家界, 2004.6: 52−61.

林雪纲： 男，1978 年生，博士生，研究方向为网络安全、生存性分析等.

许榕生： 男，1947 年生，研究员，博士生导师，主要研究方向为网络安全、黑客防范等.

熊 华： 女，1972 年生，博士后，主要研究方向为网络安全、网络对抗等.

朱淼良： 男，1946 年生，教授，博士生导师，主要研究方向为人工智能、网络安全等.