

# 基于 QoS 本体的 Web 服务描述和选择机制<sup>\*</sup>

高亚春 张为群

(西南大学计算机与信息科学学院 重庆 400715)

**摘要** 随着 Web 服务数量的急剧增长,如何从大量功能相似的 Web 服务中选择具有最好的服务质量(Quality of Service, QoS)的 Web 服务成为研究的热点。提出了一种基于 QoS 本体支持 QoS 的服务选择方法,用 QoS 本体以及相应的词汇来描述 QoS 数据,基于以上的工作,提出了一种支持 QoS 的选择机制,将 QoS 参数的语义匹配度和数值匹配度相结合,利用归一化算法对不同类型的 QoS 参数取值进行标准化,最后提出了一个公平的,动态的选择机制和相关实验结论。

**关键词** 服务质量, Web 服务, 本体, 选择机制

## Web Service Description and Selection Mechanism Based on QoS Ontology

GAO Ya-chun ZHANG Wei-qun

(College of Computer and Information Science, Southwest University, Chongqing 400715, China)

**Abstract** With the fast development of Web service, how to select the most suitable Web service within a large amount of function-equal Web services is a hot issue. Proposed a QoS ontology-based, QoS-aware selection of service. Initially specified a QoS ontology and its vocabulary to describe QoS metrics. Based on that, proposed a QoS selection model which combines semantic match and value match together. Using an optimum normalization algorithm, we presented a fair and dynamic selection mechanism and there was a demo case in the end.

**Keywords** QoS, Web service, Ontology, Selection mechanism

### 1 引言

随着因特网的发展,基于 Web 的应用数量以惊人的速度增长着。在 Web 服务出现以前,由于各个组织、机构之间平台的互异性,分布式网络面临的一个重大问题是各种平台之间的互操作性太差。Web 服务标准(WSDL, UDDI 和 SOAP)的出现解决了这个问题,但是这种 Web 服务标准对服务的描述能力非常有限,特别是缺乏对服务的语义描述,难以在逻辑层面上实现基于语义的服务发现、服务流程组合和服务动态调用。针对这样的问题,人们将语义 Web 技术和 Web 服务进行结合,产生了语义 Web 服务即 SWS(Semantic Web Service)。

现有的语义 Web 服务发现方法主要是从服务的功能匹配出发,而缺乏对服务质量 QoS(Quality of Service)的语义描述,所以难以从功能相同的众多服务中为用户选择最佳的服务。由于 QoS 对于 Web 服务在商业领域的成功应用非常关键。因此,如何提供具有 QoS 保证的 Web 服务正在引起人们的关注<sup>[1]</sup>。目前,支持 QoS 的语义 Web 服务的研究越来越多,主要是通过引入 QoS,比较服务提供者和请求者的 QoS 数值,在功能匹配的基础上进一步筛选,从而找到最匹配的服务,但是上述的方法通常只考虑了 QoS 的数值匹配,没有考虑到 QoS 参数的语义匹配度。

针对上述的问题,本文提出一种支持 QoS 的语义 Web 服务选择模型,该模型基于 QoS 本体以及相应的词汇,提出一种将语义匹配度和数值匹配度相结合,根据组合匹配度来

选择服务的方法,该方法在进行语义匹配的基础上再进行数值匹配,在 Web 服务的查全率和查准率上都有所改进。

### 2 QoS 本体

本体(Ontology)在哲学领域称为“存在论”,在计算机科学和人工智能领域则译为“本体”。施图德等在格鲁伯和波尔斯特定义的基础上,对 ontology 给出了一个最完整的定义:“ontology 是对概念体系的明确的、形式化的、可共享的规范”<sup>[2]</sup>。通俗的讲,本体就是用来描述某个领域概念以及概念之间的联系,使得这些概念和联系在共享的范围内有着明确唯一的定义,达成一种共识,这样人、机器之间就可以进行交流。因为本体具有良好的概念层次结构和对逻辑推理的支持,在查准率和查全率上有更好的保证,因而在 Web 服务的发现和匹配过程中得到了广泛的应用。

鉴于 OWL 在本体描述方面的优越性,本文采用 OWL 作为本体建模语言,并使用 Protégé 作为本体的编辑工具,制定了 Web 服务的 QoS 本体模型。

前人对 QoS 做出了不同的定义,著名学者 Jorge Cardoso 指导性地给出了服务质量评价模型中应包括的因素,具体包括费用(cost)、时间(time)、可靠性(reliability)和可信性(fidelity)。在 AgFlow 系统中选择以下 5 个通用的服务属性来评价 QoS 服务:服务运行成本(execution price)、运行时间(execution duration)、信任度(reputation)、成功率(successful execution rate)和可用性(availability)。Medjahed B 提出了把服务属性分成多类,并扩大到加密、安全等属性方面<sup>[3]</sup>。综合

<sup>\*</sup> 本文受到重庆市自然科学基金重点项目“软件测试技术和方法研究”(CSTC, 2006BA2003)支持。高亚春 在读硕士研究生,研究方向为软件工程、Web 服务;张为群 教授,硕士生导师,主要研究领域为形式化软件工程、软件测试。

前人的研究成果,本文把 QoS 分为通用领域 QoS 和特定领域 QoS,通用领域 QoS 适合于任何一个 Web 服务,是一个可调整、可扩展的属性集合,特定领域 QoS 是领域内专门的属性,如一个餐馆的位置、价格等,领域属性也是一个可扩展的属性,由领域专家根据具体的服务类别设定出相关的属性,具体过程在本文中不作详细讨论。

在本文中我们将 QoS 本体分为 3 种:顶层本体、中层本体和底层本体。顶层本体用于描述最基本的质量概念,定义和质量有关的基本属性(包括对象属性和数据属性);中层本体定义通用领域 QoS 的概念;底层本体也可叫做应用层本体,用于描述特定领域的 QoS。

### 2.1 QoS 顶层本体

在顶层本体中包括的概念(类)和属性主要包括:

(1) QoSParameter:代表了在一个 Web 服务的非功能属性。

(2) Metric:这个类定义了每个 QoS 参数的值是如何分配的。每个 Metric 对象包括一个 MetricType 和一个 Value,一个参数的 MetricType 可能是文字、数字、布尔值或者字符串。Value 是一个代表了 QoS 参数具体取值的字符串, Metric 类可以通过 hasUnit 这个属性与 Unit 类联系起来,因为同一个参数的单位的多样性(比如时间的单位就包括小时、分、秒和毫秒),还需要通过 hasConversion 属性使 Unit 类和 ConversionUnit 类联系起来,在不同的单位之间进行转换。

(3) Tendency:这个类表示 QoS 参数值的变化对服务质量的影响,通过 hasTendency 这个属性与 QoSParameter 联系起来。比如响应时间越短,说明服务质量就越好,这类参数叫做成本型参数,这类参数的 hasTendency = small/low,相反,安全性当然是越高越好,这类参数叫做效益型参数,这类参数的 hasTendency = large/high。

(4) Aggregated:这个类通过 isAggregated 这个属性与 QoS 参数联系起来,表示某个 QoS 参数是由两个或者多个其他的参数组成的,比如,响应时间就由延迟和请求处理时间组成。

(5) Relationship:这个类表示一个 QoS 参数和其他参数之间的关系。

(6) Weight:这个类表示用户给 QoS 参数设置的权重。

QoS 的顶层本体如图 1 所示,通过该顶层本体,便可以对 Web 服务的各类 QoS 参数进行统一的、标准的描述。

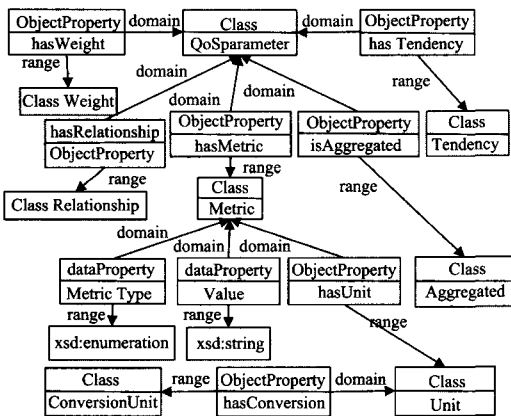


图 1 QoS 顶层本体图

### 2.2 QoS 中层本体

顶层本体主要给出了在描述一个 QoS 参数时应该描述

的内容,Web 服务的 QoS 包括哪几类参数,我们将在中层本体中进行描述。本文采用工具 protégé3. 3. 1 构造通用领域 QoS 的中层本体,综合文献[4-6]中对 QoS 的定义得出,通用领域的 QoS 参数主要包括:可用性(availability)、可靠性(reliability)、经济性(economic)、性能(performance)、可扩展性(scalability)、容量(capacity)、健壮性(robustness)、异常处理(exception handling)、准确性(accuracy)、完整性(integrity)、易使用性(accessibility)、互操作性(inter-operability)、安全性(security)、信誉(reputation)和与网络相关的 QoS 属性(network-related)。图 2 就是用 protégé3. 3. 1 构造的 QoS 中层本体图。

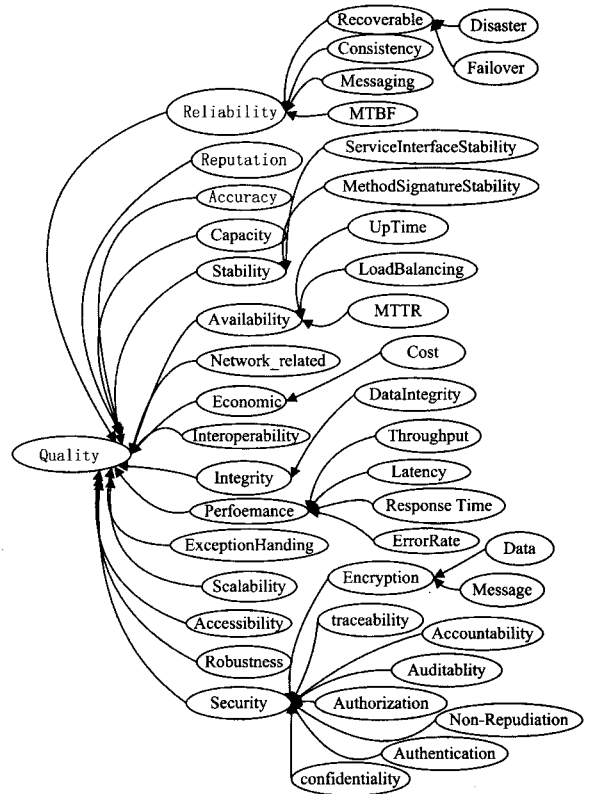


图 2 QoS 中层本体图

### 3 QoS 描述模型

为了匹配,我们提出 QoS 描述模型定义服务广告和服务请求中的 QoS 参数。现假设  $Q = \{q_1, q_2, \dots, q_i\}, i \in N, Q_i$  表示质量参数集。考虑到请求者可能没有定义服务质量的专业知识,会只基于他个人的想法提供一些质量要求,这样很容易把一些必要的质量参数在服务请求中遗漏掉,为了解决这个问题,现用  $Q_N$  表示每个服务必要的 QoS 参数集,  $Q_U$  表示用户自定义的 QoS 参数请求,那么一个完整的服务请求  $Q_R = Q_N \cup Q_U$ 。在我们的模型里面,只定义了几个质量为必须的,就是  $Q_N = \{cost, responseTime, reliability, availability, reputation\}$ ,当然,对于特定的应用系统,可以对  $Q_N$  进行扩展或改变。

这样,一组候选服务的 QoS 描述模型为:

假设通过功能匹配,已经筛选出  $x$  个具有类似功能的候选服务,我们要从其中选出具有最高服务质量的服务返回给请求者,则这些候选服务可表示为  $Q_S = \{Q_{A1}, Q_{A2}, \dots, Q_{Ax}\}$ ,其中每个服务广告的服务质量表示为:  $Q_{Ai} = ((q_{i1}, v_{i1}), (q_{i2},$

$v_{i2}), \dots, (q_{im}, v_{im})$ ), 其中,  $Q_{Ai}$  表示提供的第  $i$  个候选 Web 服务的服务质量,  $q_{ij}$  ( $1 \leq j \leq m$ ) 表示  $Q_{Ai}$  中的第  $j$  个 QoS 参数,  $v_{ij}$  表示服务提供者保证提供的相应 QoS 参数的质量。

服务请求的 QoS 描述模型:  $Q_R = ((q_1, w_1, v_1), (q_2, w_2, v_2), \dots, (q_n, w_n, v_n))$ 。其中,  $Q_R$  是一个 QoS 服务请求,  $q_k$  ( $1 \leq k \leq n$ ) 表示在服务请求中的第  $k$  个 QoS 参数名称,  $w_k$  表示请求者分配给该 QoS 参数的权值,  $v_k$  表示请求者对该 QoS 参数的期望值。

#### 4 支持 QoS 的服务选择过程

本文提出一种支持 QoS 组合相似度的服务选择算法, 从 QoS 参数语义和数值两方面计算服务请求与服务广告之间的相似程度, 从而选出最符合服务请求者质量需求的服务。

##### 4.1 QoS 参数的语义匹配

根据前面构造的 QoS 参数本体图, 当一个服务请求者提交一个服务请求时, 我们可以计算服务请求与服务广告的 QoS 参数的语义匹配度 MD(Degree of Match), 在本文中, 我们按照 Massimo Paolucci<sup>[7]</sup> 的定义把语义匹配度分为 4 个等级:

(1) 精确匹配(exact), 即服务广告中的 QoS 参数与服务请求的 QoS 参数在语义上是等价的, 即这两个概念是同一个概念;

(2) 插入匹配(plug\_in), 即服务广告中的 QoS 参数是服务请求的 QoS 参数的父概念;

(3) 包含匹配(subsume), 即服务请求的 QoS 参数是服务广告中的 QoS 参数的父概念;

(4) 失败(fail), 如果不满足以上的三种情况则匹配失败。通过增加语义匹配过程, 可以提高服务查询的查全率。

##### 4.2 数值匹配

因为不同的质量参数的值有不同的取值类型和取值范围, 为了进行数值匹配, 需要对质量参数的值进行量化和标准化。

总结下来, 质量参数的取值类型主要分为以下几种:

(1) 数字类型: 用一个数字表示, 但是有不同的取值范围;

(2) 文字类型: 由有序有限的词来表示质量参数的值, 比如, 一个服务的信誉等级就可以表示为{很高、高、中等、低、很低};

(3) 等级类型: 比如一个宾馆服务的等级可以表示为{5, 4, 3, 2, 1};

(4) 布尔值类型: 用“是”或“否”来表示。

因为要进行数值匹配, 所以首先要对非数值的取值进行量化。在我们的方法中, 我们把文字类型比如在上面所提到的用文字表示的信誉等级{很高、高、中等、低、很低}分别用 5, 4, 3, 2, 1 来表示; 对于布尔类型的取值, 我们用“1”代表是, 用“0”代表否。

因为不同的 QoS 参数的值的类型和取值范围有着很大的差别, 因此还需要对值进行标准化, 这里我们采用归一化的方法, 把值化为  $[0, 1]$  之间的数值, 在此基础上计算各候选服务的数值匹配度, 取值越接近 1, 对应的服务请求中的服务质量参数的值与服务广告中的值就越接近。

如果是离散型的数据, 标准化算法为

$$q_{ij}' = 1 - \frac{Index_{xR}}{t-1}$$

在上述公示中, 假设共有  $t$  个离散型数据,  $Index_{xR}$  是某个

离散型数据的索引位置(其中,  $Index_{xR} \in N$ , 且  $0 \leq Index_{xR} \leq (t-1)$ )。假设元素是按降序排列的, 当  $R$  在第一个位置时,  $Index_{xR} = 0$ ,  $q_{ij}' = 1$ , 当  $R$  在最后一个索引位置时,  $q_{ij}'$  的值为 0, 对于所有的  $R$ ,  $q_{ij}'$  的值均在 0 到 1 之间。

对于连续型数据, 根据它们的 hasTendency 的属性不同, 可以分为两种, 一种是成本型参数, 也就是参数的值越小, 说明服务质量越高, 如服务费用、响应时间等; 还有一类是效益性参数, 也就是参数的值越大, 说明服务质量越高, 如可靠性、安全性等。

根据参数的不同类型, 我们设定了不同的标准化算法。

(1) 如果 hasTendency = small/low, 是成本型参数, 那么标准化算法为

$$q_{ij}' = \left(1 - \frac{q_{ij} - q_{\min}}{q_{\max} - q_{\min}}\right) \quad (1)$$

(2) 如果 hasTendency = large/high, 是效益性参数, 那么标准化算法为

$$q_{ij}' = \left(1 - \frac{q_{\max} - q_{ij}}{q_{\max} - q_{\min}}\right) \quad (2)$$

其中,  $q_{\max} = \max\{q_{ij}\}$ ,  $q_{\min} = \min\{q_{ij}\}$ 。

比如, 一个用户寻找一个住宿服务, 要求价格在 \$ 100 以下, 有三个服务, 分别要价 \$ 60, \$ 80 和 \$ 90。在这个例子中因为价钱的 hasTendency 的值为 small/low, 所以在进行标准化时, 应该采用式(1)的标准化算法, 那么, 根据式(1), 这三个候选服务的价格即可分别标准化为 1, 0.333, 0。由此可见, 价格为 \$ 60 的服务最符合用户的要求。

##### 4.3 构造匹配矩阵

对于一个服务请求  $Q_R = ((q_1, w_1, v_1), (q_2, w_2, v_2), \dots, (q_n, w_n, v_n))$ , 我们假设有  $x$  个能够满足其功能需求的候选服务, 很明显, 不同的候选服务有不同的服务质量, 并且不是任何  $Q_R$  或者  $Q_{Ai}$  会有相同数量的质量参数的, 所以在构造匹配矩阵时, 我们把  $Q_R$  作为标准对每一个  $Q_{Ai}$  进行调整。这个过程包括:

(1) 针对服务请求中的  $n$  个质量参数, 计算候选服务的对应质量参数的语义匹配度 MD, 如果有  $Q_{Ai}$  中的必要质量参数的语义匹配度 MD=fail, 则说明该候选服务质量不能满足用户的请求, 将该候选服务删除, 我们假设因此而删除的服务有  $y$  个;

(2) 按照  $Q_R$  中相同的质量参数的顺序重新安排  $Q_{Ai}$  中质量参数, 并根据上述的标准化算法对  $Q_{Ai}$  中的质量参数进行标准化, 形成标准化匹配矩阵。

所以, QoS 匹配的矩阵就是  $M_Q = \{Q_{A1}, Q_{A2}, \dots, Q_{A(x-y)}\}$

$$M_Q = \begin{pmatrix} q_{11} & q_{12} & q_{13} & \dots & q_{1n} \\ q_{21} & q_{22} & q_{23} & \dots & q_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ q_{(x-y)1} & q_{(x-y)2} & q_{(x-y)3} & \dots & q_{(x-y)n} \end{pmatrix}$$

$M_Q$  是一个  $(x-y) \times n$  矩阵, 每一行就是一个服务的各质量参数的标准化数值, 每一列都对应不同服务的一个相同的质量属性。接着, 我们把用户对各质量参数分配的权值和标准化匹配矩阵相乘, 则一个服务  $Q_{Ai}$  与请求  $Q_R$  的数值匹配度为  $S_i = \sum_{j=1}^n (w_j \times M_{Q_{Ai}, j})$ , 在  $S_1, \dots, S_{(x-y)}$  中值最大的所对应的服务即为最符合服务请求的服务。按  $S_i$  的取值由大到小排列之后, 返回用户要求的服务个数, 或者根据用户设定的阈值, 返回大于阈值的  $S_i$  给用户。

## 5 模拟实验验证

为了验证基于 QoS 的组合相似度的服务选择算法的可行性和优越性,本文进行了实例数据分析,并对实验结果进行比较。

作为对比,本文还将参考文献[8,9]中的方法进行对比,它们没有计算 QoS 参数语义相似度的部分,从以下的实验结果中,我们可以看见本文的基于组合相似度的选择算法将选出更符合用户需求的服务。在本文中,我们选取必要服务参数  $Q_N$  中的费用、响应时间、可靠性、可用性和信誉构成 QoS 参数,其中,费用和响应时间的单位分别是 Dollar 和 ms,可靠性和可用性用百分数来表示,信誉等级用取值范围在[1,5]之间的整数表示。

表 1 定义了 2 个服务请求者的 QoS 请求,因为不同的服务请求者对每个参数的关注程度不同,因此对不同的服务参数分配的权重也不同。

表 1 服务请求者的 QoS 请求

	费用		响应时间		可靠性		可用性		信誉等级	
	期望 取值	权重	期望 取值	权重	期望 取值	权重	期望 取值	权重	期望 取值	权重
R1	9.30	0.3	9.62	0.1	0.88	0.2	0.75	0.2	4	0.2
R2	9.43	0.1	10.59	0.3	0.93	0.3	0.95	0.2	3	0.1

根据表 1 中定义的 QoS 向量,表 2 中提供了 5 个 Web 服务,它们的 QoS 参数值均为随机生成。

表 2 提供的 Web 服务

	费用	响应时间	可靠性	可用性	信誉等级
S1	9.32	9.23	0.95	0.93	4
S2	10.51	9.51	0.78	0.92	3
S3	9.21	10.32	0.9	0.76	5
S4	9.41	9.20	0.99	0.73	3
S5	8.99	9.89	0.93	0.96	4

首先,在本例中,因为所有的 QoS 参数都是精确匹配,所以没有需要淘汰的服务。接下来我们构造标准化匹配矩阵,通过加权操作,计算每个服务的匹配值。因为费用和响应时间是属于成本型参数,可靠性、可用性和信誉等级属于效益性参数,信誉等级属于离散型参数,根据不同的标准化算法,我们得到如表 3 所示的结果。

表 3 匹配结果

	S1	S2	S3	S4	S5	最匹配的服务
R1	0.82	0.34	0.60	0.72	0.83	S5
R2	0.83	0.44	0.39	0.72	0.71	S1

由例子我们可以看出,根据不同的服务请求,我们可以个

性化地为不同的请求者推荐不同的服务。本文与文献[8,9]最大的区别是在进行 QoS 参数语义匹配的基础上再进行数值匹配,这样,如果在精确匹配时不能满足用户的需求时,可以从插入匹配或者包含匹配中选择数值匹配度达到用户要求的 Web 服务返回给用户。

**结束语** 本文提出了一种基于 QoS 本体的 Web 服务选择方法,我们用一个 QoS 本体以及相应的词汇来描述 QoS 参数,定义了各种质量属性和它们各自的度量方法,基于以上的工作,我们提出了一个支持 QoS 的选择机制,将 QoS 参数的语义匹配度和数值匹配度相结合,利用归一化算法对不同类型的 QoS 参数进行标准化,在此基础上提出了一个公平的、动态的选择机制。本文的方法与文献[8,9]的区别在于是在进行 QoS 参数语义匹配的基础上再进行数值匹配,这样,如果在精确匹配时不能满足用户的需求时,可以从插入匹配或者包含匹配中选择数值匹配度达到用户要求的 Web 服务,语义匹配这个过程的增长为 Web 服务发现的查全率提供了更好的保证,同时,根据本文提出的标准化算法和加权过程,根据不同的请求者的要求,可以为用户提供更合适的 Web 服务。

## 参考文献

- [1] 杨胜文,史美林.一种支持 QoS 约束的 Web 服务发现模型[J]. 计算机学报,2005,28(4):589-594
- [2] Studer R, Benjamins V R, Fensel D. Knowledge Engineering: Principle and Methods[J]. Data & Knowledge Engineering, 1998,25:161-197
- [3] 李春梅,蒋运承.具有 QoS 约束的语义 Web 服务发现的研究[J]. 计算机科学,2007,34(6):116-121
- [4] Wang X, Vitvar T, Kerrigan M, et al. A QoS-aware Selection Model for Semantic Web Services [J]. Lecture Notes in Computer Science, Nov. 2006:390-401
- [5] Maximilien E M, Singh M P. A Framework and Ontology for Dynamic Web Service Selection[J]. IEEE Internet Computing, 2004(10):84-93
- [6] Kim H M, Sengupta A, Evermann J. MOQ: Web Services Ontologies for QoS and General Quality Evaluations// European Conference on Information Systems (ECIS 2005). Regensburg, Germany, May 2005
- [7] Paolucci M, Kawamura T, Payne T R, et al. Semantic Matching of Web Services Capabilities// Proceedings of the 1st International Semantic Web Conference ISWC. Sardinia, 2002:333-347
- [8] 郭得科,任彦,陈洪辉,等.一种 QoS 有保障的 Web 服务分布式发现模型. 软件学报,2006,17(11):2324-2334
- [9] 蒋运承,史忠植. QoS 驱动的主体服务匹配[J]. 小型微型计算机系统,2005,264:687-692

(上接第 243 页)

DARPA Information Survivability Conference and Exposition. 2000,2:241-255

- [5] Wang Jian, Wang Huiqiang, Zhao Guosheng. ERAS—an Emergency Response Algorithm for Survivability of Critical Services.

Computer and Computational Sciences, 2006, 2:97-100

- [6] Hiltunen M A, Schlichting R D, Ugarte C A, et al. Survivability Through Customization and Adaptability: The Cactus Approach // DARPA Information Survivability Conference and Exposition. 2000, 1:294-307