

ICS 01.140.20

A 14

C A D A L 项 目 标 准

CADAL 51501—2012

CADAL 数字图书馆在线评价标准

CADAL Online Evaluation for Digital Libraries

第一稿

2012-05-08 发布

2012-05-09 实施

CADAL 项目管理中心 发 布

目 次

前言	117
引言	118
1 范围	119
2 规范性引用文件	119
3 术语定义	119
3.1 数字图书馆	119
3.2 评价	120
3.3 指标	120
3.4 指标体系	120
3.5 在线评价	120
4 数字图书馆在线评价标准规范内容	120
4.1 评价客体或主要要素	120
4.2 评价指标体系设计原则	120
4.3 在线评价指标体系的确立	121
4.4 指标体系权重设置方法	122
4.5 指标取值规范化处理	122
4.6 综合评价计算方法	123
附录 A (规范性附录)国内外重要数字图书馆名录(略)	123
附录 B (规范性附录)在线评价算法技术实现核心代码	123
附录 C (规范性附录)录入评价功能技术实现核心代码	130
参考文献	140
表 1 数字图书馆评价系统设计指标体系	121

前 言

《CADAL 数字图书馆在线评价标准》分为以下 6 个部分：

- 第 1 部分：前言。
- 第 2 部分：引言。
- 第 3 部分：范围。
- 第 4 部分：规范性引用文件。
- 第 5 部分：术语定义。
- 第 6 部分：数字图书馆在线评价标准规范内容。

本标准的制定依据了开放式电子图书出版结构(Open eBook Publication Structure, OEBPS)规范的要求，同时参照了大学数字图书馆国际合作计划(CADAL)项目之子项目“数字图书馆评估指标体系及其支持软件研究”项目组制定的《数字图书馆评估指标体系及其支持软件研究项目报告》。

本标准由浙江大学信息资源管理研究所提出。

本标准由大学数字图书馆国际合作计划(CADAL)项目管理中心归口。

本标准起草单位：浙江大学信息资源管理研究所。

本标准起草人：叶鹰、王素芳、刘启元等。

引 言

数字图书馆一词出现于 20 世纪 90 年代,之后,电子图书馆、虚拟图书馆、多媒体图书馆、无围墙图书馆、网络图书馆等类同术语相继出现。21 世纪以来,随着数字图书馆实践的发展,人们对于数字图书馆的认识日渐清晰,并使数字图书馆成为一个快速发展的领域。

与数字图书馆的研发和建设相比,数字图书馆的评价研究和实践相对滞后。由于“数字图书馆”概念、类型的复杂性,数字图书馆评估也相对复杂,至今难有统一认识。本标准在 CADAL 子项目“数字图书馆评估指标体系及其支持软件研究”的基础上,提出适合各类型、规模各异的数字图书馆的在线评价指标体系,并规范有关技术,以利于对当前及未来数字图书馆建设发展起督促和引导作用。

CADAL 数字图书馆在线评价标准

1 范围

本标准确立了 CADAL 数字图书馆在线评价系统的评价客体及主要要素。

本标准规定了数字图书馆在线评价指标体系设计原则。

本标准确立了数字图书馆在线评价指标体系。

本标准规定了数字图书馆指标体系权重设置方法。

本标准规定了数字图书馆综合评价计算方法。

本标准适用于各类型、规模各异的数字图书馆。

2 规范性引用文件

下列文件对于本标准的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本标准。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本标准。

《图书馆数字资源统计标准和应用指南》

美国顾客满意度模型(American Customer Satisfaction Index, ACSI)

基于 5S 数字图书馆质量评价模型

ARL DigiQUAL, 可以从以下网址获得: <http://www.digiqual.org>

英国绩效评价模型 eVALUED

ARL E-metrics

欧盟 COUNTER

欧洲委员会 EQUINOX

欧盟和美国 NSF 的 DELOS

3 术语定义

数字图书馆建设中通用和 CADAL 项目界定的下列术语和定义适用于本标准。

3.1 数字图书馆 Digital Libraries

美国研究图书馆协会归纳了 20 世纪 90 年代前期流行的数字图书馆的各种定义,认为这些定义具有共性的 5 个要素是:数字图书馆不是一个实体;数字图书馆需要技术连接许多图书馆的资源;数字图书馆与信息资源的联系对用户是透明的;普遍存取是数字图书馆的目标;数字图书馆馆藏不限于传统文献,还包括没有相应印本的数字信息资源。

3.2 评价 Evaluation

评价即评判和判断,明确目标测定对象的指标或属性,为了满足某个主体要求的程度,明确其价值的过程。

3.3 指标 Indicator

指标是指示特定属性的参数,体现为量化评价对象的数据,具有明确的含义并保持含义的稳定。

3.4 指标体系 Indicator System

指标体系指由多个指标组成的归一化系统,其中每个指标具有独立的意义,而所有指标的组合或分级组合体现综合评价意图并可加权加和为一个集成数据。

3.5 在线评价 Online Evaluation

在线评价指标联机运行状态下的实时评价。

4 数字图书馆在线评价标准规范内容

4.1 评价客体或主要要素

无论国内外,目前关于数字图书馆评价模型不外乎两类:一类是单一指标评价,即从资源、技术、用户、使用效益等方面进行评价;另一类是对上述几方面进行综合评价。就综合评价看,目前数字图书馆评价标准或模型逐渐集中到数据/馆藏/资源、系统/技术、用户/使用/服务、管理这几个分层上。常见的有用户—系统分层;用户和使用—资源/馆藏—系统/技术;资源—技术—服务—管理等。本标准建议按资源、技术、服务三个层面来建立指标体系。

4.2 评价指标体系设计原则

评价指标体系设计应遵循三原则:学术性、功能性和操作性,含义如下:

(1)学术性:就是要考虑指标体系的完备性和各项指标的明晰性。在指标体系设计中,理想情况下是要系统、全面地考虑各方面因素,并用尽可能少的指标以明确的方式反映数字图书馆的主要内涵、性能以及影响因素等^①。例如,数字图书馆资源量、主页响应和检索速度、基本的检索服务等功能性指标都是必须反映的指标。同时,评价指标的概念内涵必须明确、清晰,便于理解和掌握。评价指标在时间、空间、隶属关系等方面需要有明确的范围,不能互相重复。

(2)功能性:就是要兼顾指标的可测度性和指标体系的简练性。可测度性是指评价指标能够直接度量或通过一定的量化方法间接度量。评价的可测度性必须以指标的可测度性

^① 《CADAL 数字图书馆在线评价标准》. CADAL 项目管理中心 2012-05-08 发布。

为前提,从而才能保证评价的客观和公正。简练则是在理论设计与实现功能之间取得平衡,避免出现指标冗余。

(3)操作性:要求指标能够获得并能实用。如果指标数据不能获得,评价指标体系就成了纸上谈兵;只有数据可以获得,才能保证指标体系可以操作。如果指标数据可获取性差,指标体系就不能实用。为此,采取静态和动态指标相结合方针:既用前台动态获取指标,也用后台静态录入指标,并尽量保证前者占据一定比例。从对现有成型数字图书馆调查看,前台动态可自动获取指标比较少,只有主页响应时间和检索响应速度两项比较容易实现。

4.3 在线评价指标体系的确立

在对国内外 60 多个有代表性的、已经基本成型的数字图书馆项目进行调研并确认指标可测度性的基础上,建议采用如表 1 所示之数字图书馆评价系统设计指标体系。

表 1 数字图书馆评价系统设计指标体系

一级指标	权重	二级指标	权重	三级指标	测度单位	权重	获取类型
R 资源	0.35	R1 资源数量	1/3	R11 总量(种)	NUM	1/2	A
				R12 自建库	1/0	1/2	A
		R2 资源特色	1/3	R21 古籍	1/0	1/2	A
				R22 多媒体(音频、视频、图像)	NUM/3	1/2	A
		R3 资源跨度	1/3	R31 时间跨度(2011 年为当前时间)	NUM	1/2	A
				R32 类型跨度(图书;期刊;报纸;论文;特种文献(技术报告、专利、标准);古籍(甲骨、金石);缩微;图像;音频;视频;实物/模型;其他)	NUM/12	1/2	A
T 技术	0.3	T1 信息组织	1/3	T11 标引规范	1/0	1/2	A
				T12 元数据应用	1/0	1/2	A
		T2 信息检索	1/3	T21 基本检索	1/0	1/3	A
				T22 高级检索(检索途径:书名、著者、关键词、出版机构、年限、主题词、ISBN/ISSN、语种、排序、分类、引文、其他(丛书))	NUM/12	1/3	A
				T23 特色检索(图像、视频、音频、全文、书法字、其他)	NUM/6	1/3	A
		T3 响应速度	1/3	T31 主页响应	NUM	1/2	B
T32 检索响应	NUM			1/2	B		

续表

一级指标	权重	二级指标	权重	三级指标	测度单位	权重	获取类型
S 服务	0.35	S1 基本服务	1/2	S11 浏览导航	1/0	1/3	A
				S12 在线咨询	1/0	1/3	A
				S13 用户指南(FAQ/HELP/FEEDBACK)	NUM/3	1/3	A
		S2 特色服务	1/2	S21 个性化(My Library、个性收藏、个性空间)	1/0	1/3	A
				S22 2.0(应用 RSS、BLOG、Wiki 等 Web2.0 技术)	1/0	1/3	A
				S23 用户权益(用户隐私、免费、特殊用户(儿童、残疾人等)、其他)	NUM/4	1/3	A

$$\text{Score} = \text{Value} / \text{Max} \times 100$$

注：R12、R21、T11、T12、T21、S11、S12、S21、S22 采用判别有为 1、无为 0 的取值方式；NUM 表示数量值，取值范围各不相同；R11 资源总量的单位，取种册，或者记录数(如中国期刊网、万方、维普、ACM 数字图书馆等全文数据库类型的数字图书馆)；R31 时间跨度取值为“2011—最早时间”；主页响应和检索响应是以“10000—响应时间”处理，超时时间设置为 10000 ms，此种情况一律设置 1000 值；数据获取类型：A 表后台存储静态调用；B 表前台即时动态获取。

4.4 指标体系权重设置方法

目前数字图书馆评价中指标权重的确定方法大致可分为两类：主观赋权法和客观赋权法。前者是评价者根据经验主观判断或根据对各指标的主观重视程度进行赋权的方法，如专家调查法、二项系数法、层次分析法(AHP 法)等；后者是通过建立一定的数学模型计算出权重系数的方法，如主成分分析法、熵技术法、均方差法及目标规划法等。无论哪种方法都有其优劣之处。主观赋权法随意性较大，评价准确性和可靠性较差，但指标的相对重要程度一般不会违反人们的常识；客观赋权法存在赋权的客观标准，通过计算得出评价指标的权重系数，而非人为给定，但其缺陷是，有时计算结果无法解释。

为兼顾对资源、技术、服务侧重不同的评价，本标准建议采取推荐赋权和开放赋权相结合的方式。用专家给出的赋权作为前台系统默认指标权重，其中，一级指标资源 R、技术 T、服务 S 权重系数分别赋予 0.35、0.3、0.35，二级和三级指标权重系数一律采用平均赋权的方式(具体见表 1)。同时，在后台系统软件中设置用户自由赋权功能，允许用户或评价者在实际操作中根据评价需求自行调整或设定权重。

4.5 指标取值规范化处理

为消除指标体系中某个指标取值过大或过小对于数字图书馆总体评价和排名造成的极端影响，需对指标取值进行规范化处理，即某一指标的分值等于该指标原始采集数据值除以所有指标原始采集数据值中最大者再乘以 100，即 $\text{Value} = 100 \times \text{Value}_i / \text{Max}(\text{Value}_i)$ ($i=1, 2, 3, \dots, n$)。

4.6 综合评价计算方法

数字图书馆总体评价模型是通过计算每个数字图书馆在资源、技术和服务三方面所得分值总和的方式进行,具体计算过程如下述公式(1)(2)(3)所示。

数字图书馆总体得分计算公式为:

$$\text{Total} = rR + tT + sS \quad \text{公式(1)}$$

其中, Total 表示某个数字图书馆的总体得分; R 表示一级指标“资源”得分, r 为其权重, 初始值为 0.35; T 表示一级指标“技术”得分, t 为其权重, 初始值 0.30; S 表示一级指标“服务”得分, s 为其权重, 初始值为 0.35。

一级和二级指标值 R/T/S 计算公式为:

$$\text{Score}_i = W_{i1} \times F_{i1} + W_{i2} \times F_{i2} (+ W_{i2} \times F_{i2}) \quad \text{公式(2)}$$

其中, Score_i 表示相应第 i 个二级指标或一级指标的计算所得分值; F_i 为指标值; W_{i2} 表示第 i 个二级或三级指标 F_{i2} 的权值; $(+ W_{i2} \times F_{i2})$ 在当二级指标拥有三个三级指标的时候使用, 按照平均赋值原则, $W_{i2} = W_{i2} (= W_{i2}) = 1/3$ (3)

三级指标值 F_i 计算公式为:

$$F_i = 100 \times V_i / \text{Max}(V_{i1}, V_{i2}, \dots, V_{ik}, \dots, V_{in}) \quad \text{公式(3)}$$

其中, F_i 表示相应第 i 个三级指标的计算所得分值, 结果保留两位小数; V_i 表示相应第 i 个指标的原始采集数据; n 表示数字图书馆总数; $\text{Max}(V_{i1}, V_{i2}, \dots, V_{ik}, \dots, V_{in})$ 表示对应于相应指标的最大原始数据值。 F_i 是研究者对指标原始采集值规范化处理后的值。

上述操作适用于各类型、规模各异的数字图书馆。

附 录 A

(规范性附录)

国内外重要数字图书馆名录

(略)

附 录 B

(规范性附录)

在线评价算法技术实现核心代码

```
using System;
using System. Collections. Generic;
using System. Linq;
using System. Web;
using System. Web. UI;
using System. Web. UI. WebControls;
using System. Data. SqlClient;
using System. Data;
```

CADAL 项目标准规范汇编(四)

CADAL 51501—2012

```
using System. Configuration;
using System. Net;
using System. IO;
using System. Threading;
using Liuqiyuan;

public partial class admin_processing: System. Web. UI. Page
{
    //声明变量
    string connStr = Convert. ToString(ConfigurationManager. ConnectionStrings
["ASDLConnStr"]);
    int jilushu;
    protected int ID;
    protected string name;
    protected string type_name;
    protected string URL;
    protected string description;
    protected double Total;
    protected double R;
    protected double R1;
    protected double R2;
    protected double R3;
    protected double R11;
    protected double R11_S;
    protected double R12;
    protected double R12_S;
    protected double R21;
    protected double R21_S;
    protected double R22;
    protected double R22_S;
    protected double R31;
    protected double R31_S;
    protected double R32;
    protected double R32_S;
    protected double T;
    protected double T1;
    protected double T2;
    protected double T3;
    protected double T11;
```

protected double T11_S;
protected double T12;
protected double T12_S;
protected double T21;
protected double T21_S;
protected double T22;
protected double T22_S;
protected double T23;
protected double T23_S;
protected double T31;
protected double T31_S;
protected double T32;
protected double T32_S;
protected double S;
protected double S1;
protected double S2;
protected double S11;
protected double S11_S;
protected double S12;
protected double S12_S;
protected double S13;
protected double S13_S;
protected double S21;
protected double S21_S;
protected double S22;
protected double S22_S;
protected double S23;
protected double S23_S;
protected double M_R11;
protected double M_R12;
protected double M_R21;
protected double M_R22;
protected double M_R31;
protected double M_R32;
protected double M_T11;
protected double M_T12;
protected double M_T21;
protected double M_T22;
protected double M_T23;

CADAL 项目标准规范汇编(四)

CADAL 51501—2012

```
protected double M_T31;  
protected double M_T32;  
protected double M_S11;  
protected double M_S12;  
protected double M_S13;  
protected double M_S21;  
protected double M_S22;  
protected double M_S23;
```

//dataTable 自动处理程序

```
public DataTable dt_All(string cmdStr, string TabelName)
```

```
{  
    SqlConnection conn=new SqlConnection(connStr);  
    SqlDataAdapter sda=new SqlDataAdapter(cmdStr, conn);  
    DataSet ds_All=new DataSet();  
    sda. Fill(ds_All, TabelName);  
    return ds_All. Tables[TabelName];  
}
```

```
protected void Page_Load(object sender, EventArgs e)
```

```
{  
    AdminUse admin_Use=new AdminUse();  
    jilushu=admin_Use. count_Rows_Index+1;  
}
```

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{  
    try  
    {  
        for (ID=1; ID < jilushu; ID++)  
        {  
            //提取最大值 布尔值不能查出 按 1 即可  
            string cmdStr_Max="SELECT MAX(R11) AS M_R11, MAX  
(R12) AS M_R12, MAX(R22) AS M_R22, MAX(R31) AS M_R31,  
MAX(R32) AS M_R32, MAX(T22) AS M_T22, MAX(T23) AS M_  
T23, MAX(T31) AS M_T31, MAX(T32) AS M_T32, MAX(S13)  
AS M_S13, MAX(S23) AS M_S23 FROM [Index]";  
            DataTable dt_Max=dt_All(cmdStr_Max, "Index_Max");  
            M_R11=Convert. ToDouble(dt_Max. Rows[0]["M_R11"]);
```

```
M_R12=Convert. ToDouble(dt_Max. Rows[0]["M_R12"]);
M_R22=Convert. ToDouble(dt_Max. Rows[0]["M_R22"]);
M_R31=Convert. ToDouble(dt_Max. Rows[0]["M_R31"]);
M_R32=Convert. ToDouble(dt_Max. Rows[0]["M_R32"]);
M_T22=Convert. ToDouble(dt_Max. Rows[0]["M_T22"]);
M_T23=Convert. ToDouble(dt_Max. Rows[0]["M_T23"]);
M_T31=Convert. ToDouble(dt_Max. Rows[0]["M_T31"]);
M_T32=Convert. ToDouble(dt_Max. Rows[0]["M_T32"]);
M_S13=Convert. ToDouble(dt_Max. Rows[0]["M_S13"]);
M_S23=Convert. ToDouble(dt_Max. Rows[0]["M_S23"]);

//TextBox1. Text=Convert. ToString(M_R11);

//提取原值 布尔值 1.00
string cmdStr = "SELECT * FROM [Index] WHERE ID = "
+ID;
DataTable dt=dt_All(cmdStr, "Index");
R11=Convert. ToDouble(dt. Rows[0]["R11"]);
R12=Convert. ToDouble(dt. Rows[0]["R12"]);
R21=Convert. ToDouble(dt. Rows[0]["R21"]);
R22=Convert. ToDouble(dt. Rows[0]["R22"]);
R31=Convert. ToDouble(dt. Rows[0]["R31"]);
R32=Convert. ToDouble(dt. Rows[0]["R32"]);
T11=Convert. ToDouble(dt. Rows[0]["T11"]);
T12=Convert. ToDouble(dt. Rows[0]["T12"]);
T21=Convert. ToDouble(dt. Rows[0]["T21"]);
T22=Convert. ToDouble(dt. Rows[0]["T22"]);
T23=Convert. ToDouble(dt. Rows[0]["T23"]);
T31=Convert. ToDouble(dt. Rows[0]["T31"]);
T32=Convert. ToDouble(dt. Rows[0]["T32"]);
S11=Convert. ToDouble(dt. Rows[0]["S11"]);
S12=Convert. ToDouble(dt. Rows[0]["S12"]);
S13=Convert. ToDouble(dt. Rows[0]["S13"]);
S21=Convert. ToDouble(dt. Rows[0]["S21"]);
S22=Convert. ToDouble(dt. Rows[0]["S22"]);
S23=Convert. ToDouble(dt. Rows[0]["S23"]);

//计算三级指标(单项)分值 原值/最大值×100
R11_S=Math. Round(R11 / M_R11 * 100, 2);
```

```

R12_S=Math. Round(R12 / M_R12 * 100, 2);
R21_S=Math. Round(R21 * 100, 2); //布尔值等于原值×100
R22_S=Math. Round(R22 / M_R22 * 100, 2);
R31_S=Math. Round(R31 / M_R31 * 100, 2);
R32_S=Math. Round(R32 / M_R32 * 100, 2);
T11_S=Math. Round(T11 * 100, 2);
T12_S=Math. Round(T12 * 100, 2);
T21_S=Math. Round(T21 * 100, 2);
T22_S=Math. Round(T22 / M_T22 * 100, 2);
T23_S=Math. Round(T23 / M_T23 * 100, 2);
T31_S=Math. Round(T31 / M_T31 * 100, 2);
T32_S=Math. Round(T32 / M_T32 * 100, 2);
S11_S=Math. Round(S11 * 100, 2);
S12_S=Math. Round(S12 * 100, 2);
S13_S=Math. Round(S13 / M_S13 * 100, 2);
S21_S=Math. Round(S21 * 100, 2);
S22_S=Math. Round(S22 * 100, 2);
S23_S=Math. Round(S23 / M_S23 * 100, 2);

```

//计算二级指标和一级指标值 按均值算法

```

R1=(R11_S+R12_S) / 2;
R2=(R21_S+R22_S) / 2;
R3=(R31_S+R32_S) / 2;
T1=(T11_S+T12_S) / 2;
T2=(T21_S+T22_S+T23_S) / 3;
T3=(T31_S+T32_S) / 2;
S1=(S11_S+S12_S+S13_S) / 3;
S2=(S21_S+S22_S+S23_S) / 3;
R=(R1+R2+R3) / 3;
T=(T1+T2+T3) / 3;
S=(S1+S2) / 2;
Total=0.35 * R+0.3 * T+0.35 * S;

```

//更新入数据库

```

SqlConnection conn_upd=new SqlConnection(connStr);
SqlDataAdapter sda_upd=new SqlDataAdapter(cmdStr, conn_
upd);
DataSet ds_upd=new DataSet();
sda_upd.Fill(ds_upd, "update");

```

```
SqlCommandBuilder scb_upd = new SqlCommandBuilder (sda_upd);
DataTable dt_upd=ds_upd. Tables["update"];

dt_upd. Rows[0]["R11_S"]=R11_S;
dt_upd. Rows[0]["R12_S"]=R12_S;
dt_upd. Rows[0]["R21_S"]=R21_S;
dt_upd. Rows[0]["R22_S"]=R22_S;
dt_upd. Rows[0]["R31_S"]=R31_S;
dt_upd. Rows[0]["R32_S"]=R32_S;
dt_upd. Rows[0]["T11_S"]=T11_S;
dt_upd. Rows[0]["T12_S"]=T12_S;
dt_upd. Rows[0]["T21_S"]=T21_S;
dt_upd. Rows[0]["T22_S"]=T22_S;
dt_upd. Rows[0]["T23_S"]=T23_S;
dt_upd. Rows[0]["T31_S"]=T31_S;
dt_upd. Rows[0]["T32_S"]=T32_S;
dt_upd. Rows[0]["S11_S"]=S11_S;
dt_upd. Rows[0]["S12_S"]=S12_S;
dt_upd. Rows[0]["S13_S"]=S13_S;
dt_upd. Rows[0]["S21_S"]=S21_S;
dt_upd. Rows[0]["S22_S"]=S22_S;
dt_upd. Rows[0]["S23_S"]=S23_S;

dt_upd. Rows[0]["R1"]=R1;
dt_upd. Rows[0]["R2"]=R2;
dt_upd. Rows[0]["R3"]=R3;
dt_upd. Rows[0]["T1"]=T1;
dt_upd. Rows[0]["T2"]=T2;
dt_upd. Rows[0]["T3"]=T3;
dt_upd. Rows[0]["S1"]=S1;
dt_upd. Rows[0]["S2"]=S2;
dt_upd. Rows[0]["R"]=R;
dt_upd. Rows[0]["T"]=T;
dt_upd. Rows[0]["S"]=S;
dt_upd. Rows[0]["total"]=Total;

sda_upd. Update(ds_upd, "update");
TextBox1. Text="更新完毕!";
```

```
        }//for
    }//try
    catch (Exception ex)
    {
        TextBox1. Text=Convert. ToString(ex. Message);
    }
}
}
```

附 录 C

(规范性附录)

录入评价功能技术实现核心代码

```
using System;
using System. Collections. Generic;
using System. Linq;
using System. Web;
using System. Web. UI;
using System. Web. UI. WebControls;
using System. Data. SqlClient;
using System. Net;
using System. IO;
using System. Data;
using System. Configuration;

public partial class EvaluationByInput: System. Web. UI. Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    string connStr = Convert. ToString(ConfigurationManager. ConnectionStrings
["ASDLConnStr"]);
    string name;
    string type_name;
    string URL_Index;
    string description;
    string URL_T31;
    string URL_T32;
```


double R11;
double R12;
double R21;
double R22;
double R31;
double R32;
double T11;
double T12;
double T21;
double T22;
double T23;
double T31;
double T32;
double S11;
double S12;
double S13;
double S21;
double S22;
double S23;
double R11_S;
double R12_S;
double R21_S;
double R22_S;
double R31_S;
double R32_S;
double T11_S;
double T12_S;
double T21_S;
double T22_S;
double T23_S;
double T31_S;
double T32_S;
double S11_S;
double S12_S;
double S13_S;
double S21_S;
double S22_S;
double S23_S;
double R1;

CADAL 项目标准规范汇编(四)

CADAL 51501—2012

```
double R2;
double R3;
double R;
double T1;
double T2;
double T3;
double T;
double S1;
double S2;
double S;
double TOTAL;
bool PreVerify; //是否已经验证输入

//最大值
protected double M_R11;
protected double M_R12;
protected double M_R21;
protected double M_R22;
protected double M_R31;
protected double M_R32;
protected double M_T11;
protected double M_T12;
protected double M_T21;
protected double M_T22;
protected double M_T23;
protected double M_T31;
protected double M_T32;
protected double M_S11;
protected double M_S12;
protected double M_S13;
protected double M_S21;
protected double M_S22;
protected double M_S23;

string Author;
string Institute;
string Email;

protected void DropDownList_type_name_SelectedIndexChanged(object sender,
```

```
EventArgs e)
{
    DropDownList_type_name. Text=name;
}
//自动获取响应速度方法
protected double getTime(string URL)
{
    try
    {
        //URL=Convert. ToString(fieldName); //获取网址
        DateTime t1=DateTime. Now;
        WebRequest request=WebRequest. Create(URL);
        WebResponse response=request. GetResponse();
        StreamReader reader=new StreamReader(response. GetResponseStream
        ( ), System. Text. Encoding. GetEncoding("gb2312"));
        reader. Close();
        reader. Dispose();
        response. Close();
        DateTime t2=DateTime. Now;
        TimeSpan t3=t2-t1;
        int TS1=Convert. ToInt32(t3. Seconds);
        int TS2=Convert. ToInt32(t3. Milliseconds);
        //得出响应时间 10000ms 减去时间间隔
        int newTime=10000-Convert. ToInt32(TS1 * 1000+TS2);
        return newTime;
    }
    catch (Exception e)
    {
        return 1000;
    }
}
//获取记录集
public DataTable dt_All(string cmdStr, string TabelName)
{
    SqlConnection conn=new SqlConnection(connStr);
    SqlDataAdapter sda=new SqlDataAdapter(cmdStr, conn);
    DataSet ds_All=new DataSet();
    sda. Fill(ds_All, TabelName);
    return ds_All. Tables[TabelName];
}
```

```

}
//获取数据库表 Index 中各字段最大值
private void getMaxFromIndex()
{
    try
    {
        string cmdStr_Max = " SELECT MAX(R11) AS M_R11, MAX
(R12) AS M_R12, MAX(R22) AS M_R22, MAX(R31) AS M_R31,
MAX(R32) AS M_R32, MAX(T22) AS M_T22, MAX(T23) AS M_
T23, MAX(T31) AS M_T31, MAX(T32) AS M_T32, MAX(S13) AS
M_S13, MAX(S23) AS M_S23 FROM [Index]";
        DataTable dt_Max=dt_All(cmdStr_Max, "Index_Max");
        M_R11=Convert. ToDouble(dt_Max. Rows[0]["M_R11"]);
        M_R12=Convert. ToDouble(dt_Max. Rows[0]["M_R12"]);
        M_R22=Convert. ToDouble(dt_Max. Rows[0]["M_R22"]);
        M_R31=Convert. ToDouble(dt_Max. Rows[0]["M_R31"]);
        M_R32=Convert. ToDouble(dt_Max. Rows[0]["M_R32"]);
        M_T22=Convert. ToDouble(dt_Max. Rows[0]["M_T22"]);
        M_T23=Convert. ToDouble(dt_Max. Rows[0]["M_T23"]);
        M_T31=Convert. ToDouble(dt_Max. Rows[0]["M_T31"]);
        M_T32=Convert. ToDouble(dt_Max. Rows[0]["M_T32"]);
        M_S13=Convert. ToDouble(dt_Max. Rows[0]["M_S13"]);
        M_S23=Convert. ToDouble(dt_Max. Rows[0]["M_S23"]);

        //计算三级指标(单项)分值 原值/最大值×100
        R11_S=Math. Round(R11 / M_R11 * 100, 2);
        R12_S=Math. Round(R12 / M_R12 * 100, 2);
        R21_S=Math. Round(R21 * 100, 2); //布尔值等于原值×100
        R22_S=Math. Round(R22 / M_R22 * 100, 2);
        R31_S=Math. Round(R31 / M_R31 * 100, 2);
        R32_S=Math. Round(R32 / M_R32 * 100, 2);
        T11_S=Math. Round(T11 * 100, 2);
        T12_S=Math. Round(T12 * 100, 2);
        T21_S=Math. Round(T21 * 100, 2);
        T22_S=Math. Round(T22 / M_T22 * 100, 2);
        T23_S=Math. Round(T23 / M_T23 * 100, 2);
        T31_S=Math. Round(T31 / M_T31 * 100, 2);
        T32_S=Math. Round(T32 / M_T32 * 100, 2);
        S11_S=Math. Round(S11 * 100, 2);
    }
}

```

```
S12_S=Math. Round(S12 * 100, 2);
S13_S=Math. Round(S13 / M_S13 * 100, 2);
S21_S=Math. Round(S21 * 100, 2);
S22_S=Math. Round(S22 * 100, 2);
S23_S=Math. Round(S23 / M_S23 * 100, 2);
}
catch (Exception e)
{
    Response. Write("<script>alert('"+ e. Message + "')"; </script
>");
}
}
//计算并存入数据库
private void CalculateAndInsertDatabase()
{
    try
    {
        //计算二级指标和一级指标值 按均值算法
        R1=(R11_S+R12_S) / 2;
        R2=(R21_S+R22_S) / 2;
        R3=(R31_S+R32_S) / 2;
        T1=(T11_S+T12_S) / 2;
        T2=(T21_S+T22_S+T23_S) / 3;
        T3=(T31_S+T32_S) / 2;
        S1=(S11_S+S12_S+S13_S) / 3;
        S2=(S21_S+S22_S+S23_S) / 3;
        R=(R1+R2+R3) / 3;
        T=(T1+T2+T3) / 3;
        S=(S1+S2) / 2;
        TOTAL=0.35 * R+0.3 * T+0.35 * S;

        //插入数据库
        SqlConnection conn=new SqlConnection(connStr);
        string cmdStr="SELECT * FROM [Input]";
        SqlDataAdapter sda=new SqlDataAdapter(cmdStr, conn);
        DataSet ds=new DataSet();
        sda. Fill(ds, "Input");
        SqlCommandBuilder scb=new SqlCommandBuilder(sda);
        DataRow newRow=ds. Tables["Input"]. NewRow();
```

CADAL 项目标准规范汇编(四)

CADAL 51501—2012

```
newRow["name"]=name;  
newRow["type_name"]=type_name;  
newRow["description"]=description;  
newRow["URL_Index"]=URL_Index;
```

```
newRow["R11"]=R11;  
newRow["R12"]=R12;  
newRow["R21"]=R21;  
newRow["R22"]=R22;  
newRow["R31"]=R31;  
newRow["R32"]=R32;  
newRow["R11_S"]=R11_S;  
newRow["R12_S"]=R12_S;  
newRow["R21_S"]=R21_S;  
newRow["R22_S"]=R22_S;  
newRow["R31_S"]=R31_S;  
newRow["R32_S"]=R32_S;
```

```
newRow["T11"]=T11;  
newRow["T12"]=T12;  
newRow["T21"]=T21;  
newRow["T22"]=T22;  
newRow["T23"]=T23;  
newRow["T31"]=T31;  
newRow["T32"]=T32;  
newRow["T11_S"]=T11_S;  
newRow["T12_S"]=T12_S;  
newRow["T21_S"]=T21_S;  
newRow["T22_S"]=T22_S;  
newRow["T23_S"]=T23_S;  
newRow["T31_S"]=T31_S;  
newRow["T32_S"]=T32_S;
```

```
newRow["URL_T31"]=URL_T31;  
newRow["URL_T32"]=URL_T32;  
newRow["S11"]=S11;  
newRow["S12"]=S12;  
newRow["S13"]=S13;  
newRow["S21"]=S21;
```

```
newRow["S22"]=S22;
newRow["S23"]=S23;
newRow["S11_S"]=S11_S;
newRow["S12_S"]=S12_S;
newRow["S13_S"]=S13_S;
newRow["S21_S"]=S21_S;
newRow["S22_S"]=S22_S;
newRow["S23_S"]=S23_S;

newRow["R1"]=R1;
newRow["R2"]=R2;
newRow["R3"]=R3;
newRow["T1"]=T1;
newRow["T2"]=T2;
newRow["T3"]=T3;
newRow["S1"]=S1;
newRow["S2"]=S2;
newRow["R"]=R;
newRow["T"]=T;
newRow["S"]=S;
newRow["TOTAL"]=TOTAL;

newRow["Author"]=Author;
newRow["Institute"]=Institute;
newRow["Email"]=Email;

ds. Tables["Input"]. Rows. Add(newRow);
sda. Update(ds, "Input");
conn. Close();
}
catch(Exception e)
{
    Response. Write("<script>alert('"+e. Message+"'"); </script
>");
}
}
protected void ButtonVerify_Click(object sender, EventArgs e)
{
    try
```

```
{
    name=Textbox_name. Text. Trim();
    URL_Index=Textbox_URL_Index. Text. Trim();
    type_name=DropDownList_type_name. SelectedValue;
    description=TextBox_description. Text. Trim();
    R11=double. Parse(TextBox_R11. Text. Trim());
    if (CheckBox_R12. Checked)
    {
        R12=1;
    }
    if (CheckBox_R21. Checked)
    {
        R21=1;
    }
    R22=double. Parse(DropDownList_R22. SelectedValue);
    R31=double. Parse(TextBox_R31. Text. Trim());
    R32=double. Parse(DropDownList_R32. SelectedValue);
    if (CheckBox_T11. Checked)
    {
        T11=1;
    }
    if (CheckBox_T12. Checked)
    {
        T12=1;
    }
    if (CheckBox_T21. Checked)
    {
        T21=1;
    }
    T22=double. Parse(DropDownList_T22. SelectedValue);
    T23=double. Parse(DropDownList_T23. SelectedValue);
    URL_T31=TextBox_T31. Text. Trim();
    URL_T32=TextBox_T32. Text. Trim();
    T31=getTime(URL_T31);
    T32=getTime(URL_T32);

    if (CheckBox_S11. Checked)
    {
        S11=1;
    }
}
```



```
    }
    if (CheckBox_S12. Checked)
    {
        S12=1;
    }
    S13=double. Parse(DropDownList_S13. SelectedValue);
    if (CheckBox_S21. Checked)
    {
        S21=1;
    }
    if (CheckBox_S22. Checked)
    {
        S22=1;
    }
    S23=double. Parse(DropDownList_S23. SelectedValue);

    Author=TextBox_Customer. Text. Trim();
    Institute=TextBox_Institute. Text. Trim();
    Email=TextBox_Email. Text. Trim();

    getMaxFromIndex();
    CalculateAndInsertDatabase();

    Response. Redirect("EvaluationByInputResult. aspx? name=" +
name + "&URL_Index=" + URL_Index + "&TOTAL=" + Math.
Round(TOTAL, 2) + "&Author=" + Author);
}
catch (Exception ex)
{
    Response. Write("<script>alert('+"系统错误, 请审核您输入的
数据是否正确!"+'); </script>");
}
//Response. Write(name+URL_Index+type_name+R11+"... R12"+R12+
"... R32="+R32+"... R22="+R22+"... T12="+T12+"... T22="+T22+"...
T32="+T32+"... S22="+S22+"... S12="+S12+"... S22="+S22);
}
}
```

参 考 文 献

- [1] Saracevic T. Digital library evaluation: toward an evolution of concepts[J]. Library Trends, 2000(2): 350-369.
- [2] Ying Zhang. Developing a Holistic Framework for Digital Library Evaluation [D]. Dissertation for PhD. of the State University of New Jersey, 2007.
- [3] Fuhr N. Evaluation of digital libraries[J]. International Journal of Digital library, 2007, 8(1): 21-38.
- [4] Gary Marchionini. Evaluating digital libraries: a longitudinal and multifaceted view[J]. Library Trends, 2000, 49(2): 307.
- [5] Jeng H. J. Usability of the Digital Library: an Evaluation Model [D]. Dissertation for PhD. of the State University of New Jersey, 2006.
- [6] Lesk M. Practical Digital Library: Books, Bytes and Bucks[M]. San Francisco: Morgan Kaufmann Publishers, 1997.
- [7] 王居平. 数字图书馆评价的理论和方法[M]. 合肥: 安徽大学出版社, 2008.
- [8] 吴建华. 数字图书馆评价方法[M]. 北京: 科学出版社, 2009.
- [9] 王海娟. 数字图书馆评价指标体系初探[J]. 图书馆工作与研究, 2004(1): 13-17.
- [10] 刘炜. 国际大都市图书馆数字图书馆建设评估研究[J]. 图书情报工作, 2007(5).
- [11] 杨谦. 数字图书馆评价指标构成要素研究[J]. 情报科学, 2002(12): 1302-1304.