

基于 DSpace 构建机构仓储的本地化实践

陈 和 萧德洪 林丽敏

(厦门大学图书馆 厦门 361005)

[摘要] 对开源软件 DSpace 系统进行简单介绍的基础上, 以厦门大学运用 DSpace 系统构建机构仓储为例, 对 DSpace 系统中的几个主要有关界面显示的系统文件进行中文化处理, 同时对系统的界面与功能进行调整和优化, 另外对系统运行所需的邮件服务器也作了一些改进, 从而使其更符合国内有关机构或单位的实际应用需求和用户使用习惯。

[关键词] DSpace 机构仓储 本地化

[分类号] TP39

Localization Practice of Institutional Repository Base on DSpace

Chen He Xiao Dehong Lin Limin

(The Xiamen University Library, Xiamen, 361005, China)

[Abstract] Base on the introduction of the open source DSpace software, many useful steps are introduced to accord with the really application requirement of internal correlative institution and to adapt for the internal user' s usage habit. The steps includes several system files of DSpace about language be Chinese localized, the system interfaces be adjusted, the system functions be optimized, and the mail server' s function be improved. All the steps are carried out on Xiamen University Institutional Repository which was built with DSpace.

[Keywords] DSpace Institutional Repository Localization

1 引言

DSpace系统是由美国麻省理工学院图书馆与惠普实验室共同开发, 于2002年11月根据BSD开放源代码许可协议发布的开放源代码数字机构仓储系统。DSpace系统因其强大的功能和实用性, 在全球的不同大学和研所得得到广泛的运用。厦门大学图书馆出于长期保存管理学校的教学和科研学术成果、促进学术交流、融入国际学术开放获取(Open Access)运动等目的, 于2005年下半年开始着手研究机构仓储系统, 并最终选择开放源代码DSpace软件作为构建机构仓储的软件平台^[1], 并于次年8月正式投入运行。在此, 仅以基于DSpace构建的厦门大学机构仓储的本地化实践作一个简单介绍。

2 DSpace 系统概述

DSpace系统主要用于长期保存管理教育科研机构的各种数字化资源, 包括论文、图书、数据集、学习资源、图像、乐谱、预印本、录音记录、技术报告、视频、工作文档等, 几乎包括了所有格式的数字化的文字、图像、音频和视频资源^[2]。DSpace系统采用灵活的位(bit)存储和功能存储方式, 实现上述资源的长期存储。DSpace系统中的数据描述默认采用DC

(Dublin Core) 元数据, 同时该系统支持 OAI-PMH 2.0 协议, 为存储其中的数据进行存储、交换、收割提供了一个数据标准和平台。另外, DSpace 系统在数据组织、数字资源的摄入与导出、 workflow 管理、用户与数字对象的管理等等都有强大而实用的功能^[1]。然而, 当 DSpace 系统具体应用于某一领域或一个机构时, 还有许多不足之处, 还需要针对具体情况和应用需要作适当的修改或改进, 即需要一个本地化的过程, 包括系统界面语言的中文化, 系统界面调整, 系统功能优化, 以及有关功能服务的改进等等。

3 系统软件界面汉化

DSpace 系统为国外学者开发的开源软件, 所以无论其所使用的界面语言还是系统使用习惯, 都沿用了国外(美国)的方式。因此, 原 DSpace 系统展现在国人面前的都是英文界面的, 这对于习惯于使用中文界面的用户以及英文水平有待提高的用户来说, 使用起来并不方便, 因此, 本地化的第一步, 就要实现软件界面的汉化。系统汉化并不是简单地进行文件翻译, 它涉及到编码环境以及相关文件的编码设置等。

3.1 统一编码环境^[3]

在进行汉化之前以及继续下述操作前一个重要的条件是: 统一编码环境。首先在数据库端, 安装 PostgreSQL 数据库时必须用命令 `./configure --enable-multibyte --enable-unicode --with-java`; 在安装 Oracle 数据库时, 不要采用默认的编码方式, 而是选择 UTF-8 编码, 否则在以后运用中可能会出现乱码现象。然后修改 Tomcat 服务器的 `[tomcat]/conf/server.xml` 文件中的代码, 在代码块 `<Connector port="8080"...` 中添加一行代码: `URIEncoding = "UTF-8"`。

3.2 Messages.properties 文件的汉化

DSpace 系统设计者在设计系统时就充分考虑了系统在全球化使用时可能会遇到的语言障碍问题, 他们把与用户界面有关的 JSP/Servlet 中的文本显示信息从程序中分离出来, 并重新生成一个文件, 该文件名称是 `Messages.properties`, 放在源文件目录 `[source]/config/language-packs/` 下。该文件是系统汉化的主要对象。

汉化的方法是: 首先使用文本编辑器如 EditPlus 打开此文件, 可以发现该文件结构很简单, 是由一行行 “Key=Value” 形式语句组成的文本。然后逐行把 Value 部分翻译成中文, 并以 UTF-8 编码方式保存。之后, 将该文件放到 Java 运行环境下, 使用 Java 开发工具包 (JDK) 中的 `native2ascii` 工具, 对该文件重新进行编码。最后用编码后的文件覆盖原来的文件。

翻译 `Messages.properties` 文件中 Value 值时需要注意:

仅翻译需要显示在页面上的英文即可, 对于一些 HTML 语言和某些代码, 如 “`<small>`”、“` `”、“`{0}-{1}`”、“`{2}`” 之类, 是为网页显示效果和 Java 文件专用, 不能随便更改。对于某些 Value 的翻译可以视具体情况而定。也就是说某些 Value 值要采取意译, 例如: 单词 `Community` 和 `Collection`, 直译为社区/社群和合集。厦门大学机构仓储是以各个院系为单位建立社区的, 其下又设置了若干个专题, 为了不给用户造成不必要的误解或陌生感, 把 `Community` 直接翻译成 “院系”, 把 `Collection` 翻译成 “专题”, 这样就显得直观明了。另外, 一些导航语句、提示性及警告性语句也不必直译, 可以采用比较人性化的语句来进行翻译, 由于篇幅原因, 在此不再举例。

总之, `Messages.properties` 文件中的信息都是显示在系统界面上的, 是与用户交互的 “语言”。在对 `Messages.properties` 文件进行翻译时在不偏离原文主旨的情况下, 要尽可能地贴近中国人的习惯, 多一些人性化用语, 营造使用者与系统交互时的和谐氛围。

3.3 Input-forms.xml 文件的汉化

DSpace 系统中条目提交要经过交互七个窗口界面, 其中前两个窗口界面主要是条目元数据方面的, 这些元数据的显示信息以及一些提示性语句都是放在 `[source]/config/`

input-forms.xml 文件中。翻译该文件时只要对该文件如下 Tag 中的语句进行翻译即可。翻译中注意事项同 Messages.properties 文件。

```
<label> </label>
<hint> </hint>
<required></required>
<displayed-value> </displayed-value>
```

3.4 邮件模板文件的汉化

DSPACE 系统与用户交互的方式除了系统窗口平台之间的交互外，还有一部分是通过邮件完成的，比如新用户注册、更改密码、提交信息反馈、意见反馈等。DSPACE 系统预先为这些交互过程中用到的邮件设置了模板，在需要使用到时，再把相关信息插入到模板中，然后发送到用户邮箱中。这些模板文件都被放在[source]/config/emails/目录下，而且内容都是英文的。翻译这些模板文件的方法也是用文本编辑器如 EditPlus 打开，把其中的英文内容翻译为中文。需要注意的是，“Subject:”和类似“{0}”、“{1}”的词语不要修改；其他的语句在不偏离内容主旨的原则下，可以任意发挥，因为是要给用户查看的，所以需要告诉或提示用户的内容都可以在此加上。

还有一个关键的是保存文件时请选择 UTF-8 编码方式保存，不然，当这些邮件发送到用户邮箱时可能成了乱码。在某些电子邮件系统，如亿邮电子邮件系统，可能解码方面的问题，在阅读即使使用 UTF-8 编码方式保存后的邮件时也会出现乱码现象，但是若用 Foxmail 收取该服务器上的邮件，则可以正常阅读。

3.5 帮助文件及其他文件的汉化

DSPACE 系统为了使用户更好地使用好系统，为用户提供了三种帮助文档，分别为 DSPACE 帮助文档 (DSPACE Help)、合集管理员帮助文档 (Collection Administrator Help) 和站点管理员帮助文档 (Site Administrator Help)，由于这三个文档是 HTML 编码格式，只要对其直接进行翻译即可。

提交条目上传许可文件[source]/config/default.license，是 DSPACE 系统预先默认设置的许可文件，可以对其进行直接翻译或者在其基础修改或替换为想要的文件上传许可协议。

4 界面调整与功能优化

原 DSPACE 系统界面已经显得干净、简洁，但是为了达到我们的宣传和实际使用目的，我们还是对系统的某些界面以及一些功能作了一些调整和优化。

4.1 系统首页的调整与优化

系统首页是由多个文件组成的，其结构如下图 1 所示，①区和⑤区分别是页面的天头和地脚，②、③、④区是页面的主要部分。下面对各区的调整和所在文件的修改分别阐述。

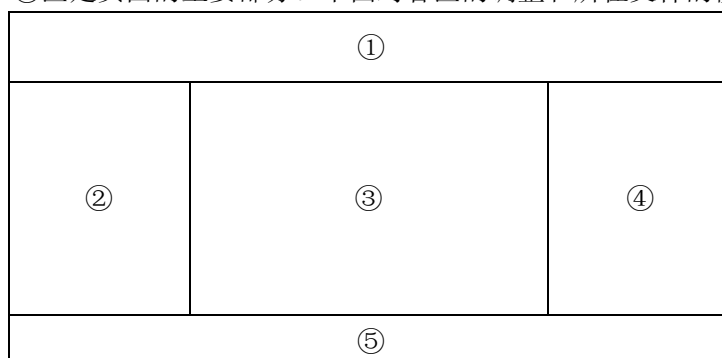


图1 DSpace 系统首页示意图

天头①区部分包括了 DSpace 系统 Logo 图片和两个链接（机构仓储首页和 DSpace 官方网站）。为达到我们的宣传效果和体现构建机构仓储的主题，我们根据原 Logo 图片的尺寸重新设计了新的 Logo 图片，对作为网页过渡用的图片 stripe.gif 也进行了一些处理。图片设计好后，需要对网页[source]/jsp/layout/header-default.jsp 进行必要的修改以正确显示图片。另外，把 DSpace 官方网站的链接（About DSpace Software）也作了屏蔽处理。

构成页面主体部分的②区，是系统页面的导航区，包括简单检索区（Simple Search）、浏览方式链接区（Browse）、工作区（Sign on to）、帮助链接区（Help）。我们主要对工作区作了一些调整。

要提交资料或管理 DSpace 系统，用户首先要在系统上注册。新用户要在原 DSpace 系统上注册，需要经过多次点击窗口才能找到注册页面。为了简化操作上的繁琐，并使之出现在显眼的位置，我们专门修改构成页面②区的[source]/jsp/layout/navbar-default.jsp 文件，在其页面代码中加入如下代码：

```
<td>.gif" width="16" height="16"/></td>  
<td nowrap="nowrap" class="navigationBarItem"><a href="<%= request.getContextPath() %>/register"><fmt:message key="jsp.layout.navbar-default.register"/></a></td>
```

同时，在 Messages.properties 文件中加入“jsp.layout.navbar-default.register = 新用户注册”。再经过适当的调整，使“新用户注册”链接出现在工作区显眼的位置，方便新用户注册，而把“我的订阅（Receive email updates）”链接放在了略次的位置。在帮助链接区，考虑到用户对 DSpace 软件及官方网站兴趣不大，对“关于 DSpace”（About DSpace）的链接进行屏蔽处理。

构成页面主体部分的③区，分为上、中、下三块区域，分别为公告区、简单检索区和社群（Community）显示区。由于②区已经有了简单检索区，我们认为在③区再加一个简单检索区，显得重复，没有多大意义。于是我们修改[source]/jsp/home.jsp 文件，把简单检索部分的代码屏蔽了，只剩下公告区和社群显示区。起到简化页面，突出主旨内容（社群显示区）的作用。

地脚⑤区部分，我们也作了修改。DSpace 从 1.3.2 版本开始已经支持 W3C 的 XHTML1.0，并且在⑤区的左边显示一个徽标 Logo 图片，点击该 Logo 可以链接到 W3C Markup Validation Service 主页。由于该 Logo 图片不是本地的，每次打开网页时才从 W3C Markup Validation Service 站点下载到本地并显示，这对网页的显示速度带来负面影响。有鉴于此，我们把这部分的文件 valid-xhtml10 和图片 w3c.bmp 从 W3C Markup Validation Service 站点直接下载到服务器，并修改[source]/jsp/layout/footer-default.jsp 文件，从而加快网页的显示速度。

4.2 增加用户注册信息

DSpace 系统默认的用户注册信息只包括用户的姓名和联系电话。基于 DSpace 构建的厦门大学机构仓储的社群是以院系为单位的，某院系的用户在系统上注册后，系统管理员将该用户加入到该院系的提交群组中。同时，为了简化工作，也是对用户的信任，用户提交的条目是不需要经过审核确认而直接被发布的。

然而，系统管理员不可能知道或认识这些新注册用户是哪个院系，不知道该把他们加入到哪个院系的提交群组中。而且，万一某个用户提交了不该提交的条目后，追查起来也不容易。针对这种情况，我们在用户注册信息中加入了“职称或身份”、“所在院系”、“其他信息”等信息栏，并要求用户在注册时如实填写相关信息。为了实现上述目的，需要对下面的文件逐一进行修改：

(1) [source]/jsp/register/profile-form.jsp 文件中主要增加如下代码:

```
post = epersonForm.getMetadata("post");
institution = epersonForm.getMetadata("institution");
info1 = epersonForm.getMetadata("info1");

<td align="right" class="standard"><label for="tpost"><strong><fmt:message
key="jsp.register.profile-form.post.field"/></strong></label></td>
<td class="standard"><input type="text" name="post" id="tpost" size="40" value="<%= post %>"/></td>
<td align="right" class="standard"><label for="tinstitution"><strong><fmt:message
key="jsp.register.profile-form.institution.field"/></strong></label></td>
<td class="standard"><input type="text" name="institution" id="tinstitution" size="40" value="<%=
institution %>"/></td>
<td align="right" class="standard"><label for="tinfo1"><strong><fmt:message
key="jsp.register.profile-form.info1.field"/></strong></label></td>
<td class="standard"><input type="text" name="info1" id="tinfo1" size="40" value="<%= info1
%>"/></td>
```

(2) [source] /src/org/dspace/app/webui/servlet/EditProfileServlet.java 文件中主要添加如下代码:

```
eperson.setMetadata("post", post);
eperson.setMetadata("institution", institution);
eperson.setMetadata("info1", info1);
```

(3) [source]/jsp/dspace-admin/eperson-edit.jsp 文件中主要增加以下代码:

```
<label for="tpost"><fmt:message key="jsp.dspace-admin.eperson.general.post"/></label>
<input name="post" id="post" size="24" value="<%=post == null ? "" : post%>"/>
<label for="tinstitution"><fmt:message key="jsp.dspace-admin.eperson.general.institution"/>
</label>
<input name="institution" id="tinstitution" size="24" value="<%=institution == null ? "" :
institution%>"/>
<label for="tinfo1"><fmt:message key="jsp.dspace-admin.eperson.general.info1"/></label>
<input name="info1" id="tinfo1" size="24" value="<%=info1 == null ? "" : info1%>"/>
```

(4) [source] /src/org/dspace/app/webui/servlet/admin/EPersonAdminServlet.java 文件中主要添加如下代码:

```
e.setMetadata("post", request.getParameter("post").equals("") ? null:
request.getParameter("post"));
e.setMetadata("institution", request.getParameter("institution").equals("") ? null:
request.getParameter("institution"));
e.setMetadata("info1", request.getParameter("info1").equals("") ? null:
request.getParameter("info1"));
```

(5) 在 Messages.properties 文件中加入如下的 Key 和值:

```
jsp.register.profile-form.post.field = 职称或身份
jsp.dspace-admin.eperson.general.post = 职称或身份
```

```
jsp.register.profile-form.institution.field =所在院系  
jsp.dspace-admin.eperson.general.institution = 所在院系  
jsp.register.profile-form.infol.field =其他信息  
jsp.dspace-admin.eperson.general.infol = 其他信息
```

(6) 在安装好后的数据库的表 **Eperson** 表中添加三个字段，分别是“post”、“institution”和“infol”，数据类型为 **Varchar2**，并根据需要设置字段大小。

4.3 作者姓与名的调整

国外人名的书写习惯是名在前，姓在后。这一点也体现在 **DSpace** 系统中，包括用户注册信息中的用户姓名和条目作者姓名。本系统在制定条目提交规范时，要求提交人在填写条目作者姓名时，把作者姓名按国内传统习惯姓和名合在一起写入到英文习惯的名字框中，而把姓氏框留空。这样，一方面避免了显示条目作者时，名在前，姓在后的“不习惯”，另一方面也方便了检索需要，因为在使用作者作为检索字段时，国人的习惯一般是直接使用姓名合在一起作为检索词进行检索，很少有姓和名分开检索的。

5 邮件服务器的认证改进

DSpace 系统默认的用户认证功能是通过 **E-mail** 和邮件服务器来完成的。新用户注册时首先需要填写本人有效的 **E-mail**，然后系统通过邮件服务器向该 **E-mail** 发送一封带 **Token** 的 **URL** 链接地址，然后用户再根据此地址完成注册。经过系统管理员授权后的用户在提交条目资料时，如果条目成功提交，或者经过管理员审核后需要修改，系统将通过邮件服务器发一封确认信或通知信给用户，以便使用户作进一步处理。因此邮件服务器在 **DSpace** 系统运行过程中起着重要的作用。

由于 1.3.2 版本及之前版本的 **DSpace** 系统只能使用不认证权限的邮件服务器，也就是说需要专门为 **DSpace** 系统配置一个不认证权限的邮件服务器。这样不仅给系统管理员带来工作负担，也为系统安全带来隐患。能不能采用校外需要认证的邮件服务器呢？正值对 **DSpace** 系统的这一功能进行改进时，**DSpace** 系统 1.4 的 **alpha** 版本已经出来了，并且发现该版本已经修正了这方面的不足^[4]。于是直接引用新版的 **E-mail** 处理 **Java** 方法，并修改 `[source]/config/dspace.cfg` 文件，在该文件中添加如下属性代码：

```
mail.server.username = USERNAME  
mail.server.password = PASSWORD
```

通过这样修改，就不必再为 **DSpace** 系统另外配置邮件服务器了，可以使用校园外的个人免费邮件服务器，避免了上述邮件服务器的不足。

6 结束语

对上述修改过的非 **JSP** 文件，全部采用覆盖原有文件的方法，而对 **JSP** 文件，在 `[source]/jsp/` 目录下新建 **Local** 目录，在此目录下，按 **JSP** 文件原有目录结构设置该文件。例如，`[source]/jsp/register/profile-form.jsp` 文件修改后的目录结构是 `[source]/jsp/local/register/profile-form.jsp`。做完上述工作后，即可以直接运行 `ant fresh_install` 进行安装或运行 `ant update` 进行更新。

要使基于开源软件 **DSpace** 构建的机构仓储系统符合本机构实际应用需求，需要对该系统进行本地化处理。而处理的内容并不止于如上所述，还有很多内容和功能需要进行改进或修正，如条目提交时的查重处理、浏览排序、元数据提交、权限管理等等，在此，仅借此起

到抛砖引玉的作用。

参考文献

- 1 陈和. DSpace 系统与厦门大学机构存储的构建. 数字图书馆论坛, 2006, (9):61-67, 75
- 2 DSpace 官方网站 <http://dspace.org/> (Accessed Dec. 20, 2006)
- 3 DSpace System Documentation. <http://dspace.org/technology/system-docs/install.html> (Accessed Dec. 20, 2006)
- 4 SourceForge.net. <http://sourceforge.net/projects/dspace/> (Accessed Dec. 20, 2006)
- 5 厦门大学学术典藏库 <http://dspace.xmu.edu.cn/> (Accessed Dec. 20, 2006)

(作者 E-mail:xmu_chen@163.com)