

基于 Notes 的文档评阅流程的实现

蔡丽娟 邹 涛 (南京大学多媒体计算机研究所 210093)

摘要:本文简介了在 Lotus Notes 环境下一个文档评阅库的实现。

关键词:评阅 提交 评阅文档 RTF 文本域 常用笔

1. 引言

近年来,随着信息科学技术的飞速发展,实现办公事务处理自动化并逐步实现无纸化办公已成为一种不可阻挡的趋势。文档评阅是日常办公中的一个重要部分,也应顺应办公自动化的趋势,从而提高办公效率和管理水平。在参与九五省科技攻关项目“智能网络文档库”时,笔者承担了文档评阅流程的实现。本文将对此作一简要介绍。

2. 人工评阅流程

在日常工作中,人们完成一篇论文、报告、计划等后,往往希望得到别人的意见,他便需将这文章寄或送给他认为合适的人,让他们发表看法(即评阅)。在得到这些意见后,即综合各人的意见,修改完善自己的文章。这就是粗略的评阅流程。我们把送交评阅的文章称为评阅文档,把寄送文档给别人评阅的人称为文档提交者,把评阅文档的人称为文档评阅者。

评阅有两种基本方式:顺序和并行。在顺序方式下,评阅文档被依次寄送给各评阅人:第一个评阅人评阅完后传给第二个评阅人,第二个评阅人评阅完后传给第三个评阅人,如此直到文档从最后一个评阅人手中寄送给提交者。在并行方式下,文档提交者将评阅文档复印多份,分别寄送给所有指定评阅者。各评阅者接到文档后即可进行评阅工作,完成后再将文档寄送给提交者。

为防止文档在某一评阅人手中停留过长时间以至阻碍其他人工作的进行,有时会要求评阅人在一定期限内完成评阅工作,否则不断催促此评阅人或将他跳过。

3. Lotus Notes 简介及 StdR46NetLib 模板分析

Lotus Notes 群件以共享式文档库为基础,以电子消息通信为流程实施的基本手段,是目前最完善地实现实际工作流程的应用软件。Lotus Notes 在我国正引起越来越广泛的关注,被越来越多的大型企事业单位和行政机关所采用。它具有以下特点:

·高效的复制机制;

- 健全的邮件通信机制;
- 高可靠的安全体系;
- 极强的工作流机制;
- 高速的全文检索引擎功能;
- 内嵌的 Internet 功能,与 Internet 实现无缝连接;
- 多平台支持。

下面简单介绍后面所用到的 Notes 中的一些概念。

(1) LotusScript : LotusScript 是 Notes 内部集成的编程接口,是一种面向对象、兼容 Basic、跨平台的编程语言,它和 Notes 的接口是通过预定义的对象类来完成的。

(2) 公式:公式是 Notes 提供的另一个集成的编程接口,是一种类似编程语言属性的表达式。如可以将值赋予变量并使用有限的逻辑控制。公式语言与 Notes 的接口是通过调用 @function 实现的。

(3) RTF: RTF(Rich Text Format)文本域是存储大量格式文本、多媒体数据的域,它可以包含不同字体与颜色的文本和图形、附件和 OLE 对象等。

Notes Designer 提供了一些功能主模板,应用开发者可在主模板基础上加以改进,实现特定功能。在实现文档评阅库时,我们选取了 StdR46NetLib 模板。此模板本身已经提供了顺序评阅功能。用户可通过 Notes 或通用 Web 浏览器访问用该模板创建的数据库。两种方式下功能基本相同,但实现和使用方式有所不同。下面介绍的都是 Notes 客户端的情况。

用户可在用此模板创建的数据库中创建新文档,编辑完后可保存或提交评阅。提交评阅前,用户需要填写评阅者、评阅时限、逾时处理,通知原作者方式等评阅信息。提交者按下提交评阅按键后,该按键的 Script 将当前文档作为评阅文档送交评阅,仅文档作者和当前评阅者(或是具有编辑者以上存取权限者)可对其修改。Script 还创建一个新文档,将当前文档的全部内容拷贝给它,把它作为原始文档,仅文档作者(或是具有编辑者以上存取权限者)可对其修改。使原始文档成为评阅文档的答复文档,以保持二者联系。另外,此按键的 Script

向第一个评阅者发送一请求评阅的邮件,在邮件的 Body 域(是一 RTF 文本域)中创建指向评阅文档的链接。当请求评阅的邮件到达某一评阅人后,Notes 自动提示他有新邮件。评阅人打开邮件数据库即可看到此邮件,双击指向评阅文档的链接,就打开了评阅文档,评阅者按下评阅文档的编辑文档按钮即可对其进行评阅。在评阅工作完成后,评阅者按下评阅完成按钮。此按钮的 Script 判断当前评阅者是不是最后一个评阅者,如果是,只需向提交者发送评阅完成邮件;如果不是,就向下一评阅者发送请求评阅邮件(邮件处理同上),如果提交者设置每次评阅后都通知原作者,那么还要向提交者发送评阅完成邮件。当提交者收到最后一个评阅者发来的评阅完成邮件,就知道所有的评阅工作已经完成。

逾期评阅文档的处理是通过一个定时代理完成的。该代理定时查看数据库中所有评阅文档。如果评阅设置了评阅时间限制,那么判断当前评阅人是否逾期尚未评阅。如果逾期且逾期处理设置为逾期则不断催促,那么就向当前评阅者发送催促邮件。如逾期且设置逾期则跳过当前评阅者,那么就向当前评阅者发送跳过邮件,并且将评阅文档往下传。为清楚表明评阅工作的进行状况,文档中设置有评阅记录域,记录提交者名、提交时间、各评阅者名、评阅时间或被跳过时间。

4. 功能完善

由于 StdR46NetLib 所提供的功能还不能完全满足我们的实际需求,需要进一步对其修改,完善其功能。

并行评阅是实际评阅流程中的一种常用方式,仅有顺序评阅一种方式必定会制约系统的灵活性和效率,所以有必要增加并行评阅的实现。并行评阅基本思想和顺序评阅一致:将文档存为两份,原始文档和评阅文档,通过请求评阅邮件通知评阅者进行评阅工作,所有评阅者都在评阅文档上评阅、修改。我们修改了一些流程控制,譬如并行方式下文档提交者必须向所有的评阅者发送请求评阅邮件,每一评阅者完成评阅后只需简单地提示他评阅工作已经完成或通知文档提交者工作完成。对逾期评阅文档的处理也有所不同。

Notes 保证共享信息安全性的方法之一是存取控制。存取控制表(ACL)规范了什么人可以以什么方式(例如创建、读、写、删除等)访问资源。在文档评阅库中,缺省使用者的权限是作者。作者可以创建新的文档和读其他文档,但是不能修改数据库中已经存在的文档(即使是他自己的文档)。这样评阅工作将不能进行。但如果将缺省使用者的存取权限提高一级至编辑者,又使每个使用者都可修改数据库中的各文档,会使评阅功能不能正常

工作,造成混乱。实际采取的方法是使用类型为“作者”的域。“作者”域只影响在数据库中具有“作者”存取权限的用户,它允许限制用户只能编辑他们自己创建的文档以及将他们列在“作者”域的文档。在评阅文档中设置两个“作者”域,一个名为 From,保存文档作者名;一个名为 CurrentEditor,保存当前评阅者名。下面以并行方式为例,说明这两个域的使用。当用户创建新文档时,From 域就得到了文档作者名,并且不再改变,文档作者就可始终修改其所作文档。在提交者按下提交评阅按钮后,该按钮的 Script 将所有评阅者名都写入评阅文档的 CurrentEditor 域,他们就可评阅该文档了。原始文档的 From 域与评阅文档的相同,CurrentEditor 域为空,只允许文档作者编辑。各评阅者评阅先后次序不确定,但每一评阅者评阅完,按了评阅完成按钮后,Script 便将此评阅者名从评阅文档的 CurrentEditor 域中删除,他便再不能评阅该文档了。

在评阅文档中,各评阅人的意见聚集一起,不利于提交者识别、采纳。曾设想将每一评阅者的意见存为文档的一个版本,这样提交者就能很清楚地知道各评阅人的不同意见,但又不利于汇总,还必须在各版本之间频繁地转换。最后我们决定给每一评阅者分配一种颜色的“笔”,并在评阅记录域中记录各人使用的颜色。这样做的依据是:评阅者的意见主要集中在正文(Body)域中,此域是一 RTF 文本域,可以设置文本属性以使文本呈现不同颜色(使用 @Command([TextSetFontColor];[color]) 公式)。但此公式仅对调用时 RTF 文本域中光标所在位置有效,当光标游走他处时,插入字符的颜色就要受左右字符颜色的控制。为解决这一问题,我们在文档进入或退出编辑模式之后发生的 Postmodechange 事件中编写 Script,判断文档如果是从读模式切换到编辑模式,则打开常用笔(@Command([TextPermanentPen];onOff),然后根据评阅者在评阅者列表中的位置设置笔的颜色。这样就能保证光标移到 Body 域的任何位置,评阅者的“笔”的颜色都始终一致。

5. 结束语

经过以上的修改,文档评阅库同时具有了顺序评阅和并行评阅两种方式,并且能给每一个评阅人设定某一颜色的笔,使得评阅人的修改和发表的意见显得很清晰。我们在 Web 上也基本实现了上述的功能修改,但由于 HTML 的限制,无法改变文本的颜色,文档作者和所有评阅者都使用一种颜色,有待进一步改进。

(来稿时间:1998年5月)