MVC架构驱动重构实现客户端网页组成

1. 介绍

在Web应用程序中，网页通常会共享类似的内容页眉，页脚和菜单。模板用于维护这种共同的内容在一个地方，它们是分开的页面特定的内容。基于模板的Web应用程序（TWA使用这样的模板来动态生成网页页面，将页面特定的内容与模板组合在一起。例如，在图1中，三个页面是使用a生成的模板和每个页面的具体内容。

TW基本上与用户交互，使用mUlti页面应用程序模型，当用户发出请求时，a网页浏览器下载并显示新页面。如图所示图2a是关于TWA最重要的事情之一是服务器将模板组合并发送到一起并在每次请求时使用页面特定的内容。因此，该模板被冗余传输并用a渲染每个请求的整页刷新。

上述重复问题的一个解决方案是将页面组合从服务器移动到浏览器。每当用户点击超链接或提交表单，网页浏览器收到当前页面和新页面之间的区别，以及然后用差异部分更新当前页面，没有整页刷新。如图2b所示，每一个请求ReqI /和Req2 /使用部分页面刷新其相关的差异。页面PI实际上是一个页面但可以有不同的统一资源定位符（URL），取决于其状态（即，所包含的内容）。这样的网络应用程序被称为单页面应用程序（SPA）。

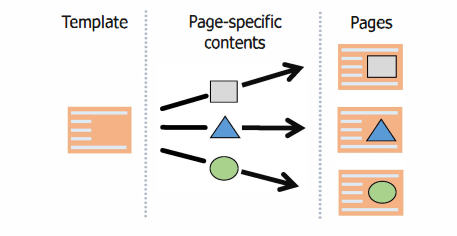


图1.基于模板的Web应用程序在Java Web应用程序中，通常使用<jsp：include>标准操作用页面特定的内容填充模板的占位符。

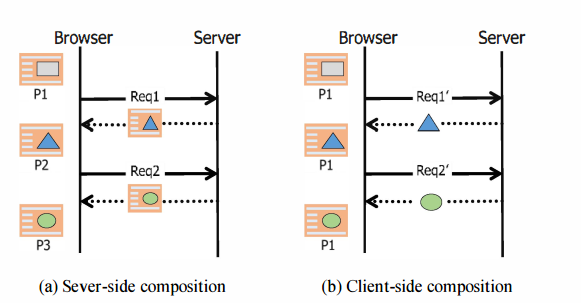


图2.网页组成

模型 - 视图 - 控制器（MVC）架构是常用于Web应用程序开发。 因此，它是自然地考虑三个组件及其相互作用作为重组时的基本分析和修改单位或将多页面应用程序重新设计为单页面应用。 但是，这些架构组件具有在以前的大多数作品中都没有被充分讨论过转型问题。

转型基本上要求其先决条件在执行之前满意。 特别是，减少了重复的代码/数据有时会导致不一致源和目标应用程序：可观察到的行为目标应用程序可能与源代码不同应用。 然而，大多数以前的研究转化问题没有指定哪里的情况转型方法无法应用。

因此，有必要指定前提条件在建筑方面进行转型的步骤MVC模式的组件。 本文提出了一种方法决定给定的整页请求是否可以重构转换成部分页面请求，而不会改变可观察的从应用程序的角度来看源应用程序的行为MVC体系结构。 还提出了重构算法描述重组步骤以及重组前提。 我们介绍重组的整体情况将算法的执行作为一个活动进行处理。另外，本文描述了我们的重要问题在实施重组算法时遇到。此外，我们介绍的案例研究结果显示当我们的工具被使用时，重组所需的努力就是减少，Web应用程序的性能得到提高。

2. 问题和背景

本文论述的重组问题可以考虑到用户界面的低级别重组（UI）结构，页面导航模型（即网络）页面之间的链接），数据模型和功能源应用程序不会更改。目标应用程序与其源应用程序有相同的外观和感觉，但它提高响应时间和数量的性能通过网络传输数据，并提供异步服务器和浏览器之间的通信能力。此外，模板和页面特定之间的耦合通过重组减少内容，所以，维护（例如，添加新的页面特定内容对于新的要求）可以更容易地执行。该选择低层次重组的理由是无缝的重构输入应用程序的多页面体系结构融入单一页面架构而不会让用户感到困惑。我们假设输入TW As是使用Java Server Pages（JSP）模型2架构其中采用MVC模式。这个假设是合理的，因为在JSP中提到了JSP模型2架构JSP的早期规范并获得了广泛接受工业。

重组问题有许多方法或将多页面应用程序重新设计为SPA。 然而，很难找到解决这个问题的研究MVC模式的角度，这是一个流行的Web应用程序开发的架构模式。 一个本文早期版本明确考虑模型，视图，控制器和它们的修改互动。 但是，以前的研究有一些限制。 重组的前提不是描述。 没有显示算法用于自动执行重组，所以实施这种方法并不容易。没有显示实验结果重组过程。

在本文中，我们改进了以前的研究并提出一个考虑重组的算法输入输出应用程序之间的一致性检查从MVC模式的角度来看。 而且，我们根据过程质量呈现案例研究的结果。

本小节简洁地描述了我们以前的工作帮助人们更多地理解本文的想法清晰。 Java Web应用程序基本上由JSP，servlet和JavaBean对象，它们映射到视图，控制器和模型，分别从角度来看MVC架构。 网页模板已用于为用户提供一致的视图并减少代码在web应用程序中。 该模板创建并管理共享组件，如菜单。 当用户提出请求时页面，请求被发送到一个servlet，它执行它商业逻辑。 执行的结果然后包含在模板来生成发送给浏览器的页面。访问JavaBeans对象来操作业务数据由请求处理期间的servlet和视图进行。 图。图3a显示了处理用户请求的动态体系结构。

该架构存在模板存在的问题下载并呈现每个用户的整页刷新请求。 因此，我们以前的工作提出了一种方法将TWA重组为客户端网页组成来解决问题。 主要想法是第一次访问应用程序时下载模板，以及那么，如果可能的话，绕过下载模板随后的访问。 另外，用户请求被转换从正常的超文本传输协议（HTTP）请求到异步JavaScript和XML（AJAX）请求在浏览器中启用部分更新。 完整的页面是完整的在JavaScript代码的帮助下在浏览器中编写更新当前的文档对象模型（DOM）树用下载的部分替换旧的页面特定的部分。图3b显示了处理用户请求的动态体系结构。因此，输出应用程序是SPA。

我们之前的工作也支持后向导航和书签功能，以使访问重组应用程序的方式与经典Web应用程序相同。 什么时候用户在SPA中点击浏览器的后退按钮，浏览器默认情况下不会回到之前的状态SPA，而是访问SPA之前访问的页面（见图4a）。 至解决问题，我们以前的工作应用了超文本标记语言5（HTML5）历史记录应用程序编程接口，其中存储当前状态浏览器历史堆栈根据用户请求并检索说明何时使用后退按钮（参见图4b）。 SPA有不同的状态，但默认情况下只有一个URL。 因此，我们的以前的工作也提出了一种分配独特的方法请求每个州的URL，以便为书签和重新访问州。

3. 重组基于模板的应用INTO单页适用

1. 重组过程

在本小节中，我们将展示我们的整体情况重组过程将TWA修改为SPA（见图5）。下面简要介绍这些步骤。

首先，工程师执行并理解来源

从用户的角度来看应用程序。然后，她收集链接和表单可以发出正常的HTTP请求重组。对于每个请求，工程师可以提取一个动态的体系结构，可以表示为一个协作图类似于图3a中的图显示Web组件之间的协作。架构可以有效地用于重组的验证步骤应用。可以实现架构的提取使用Java Web中提供的过滤器和包装器组件开发技术。提取步骤没有描述因空间有限而详细。现在，工程师了解源的功能和体系结构应用程序，并因此执行重构算法将输入的应用程序重组为SPA。验证输出申请是在重组之后进行的fmished。发现问题时，包括错过和不正确的转换，可以执行手动转换。

请注意，本文重点介绍该体系结构转型及其支持工具。 因此，我们已经发展一个Eclipse插件来支持“执行重组”算法“步骤，如下所述部分详细。

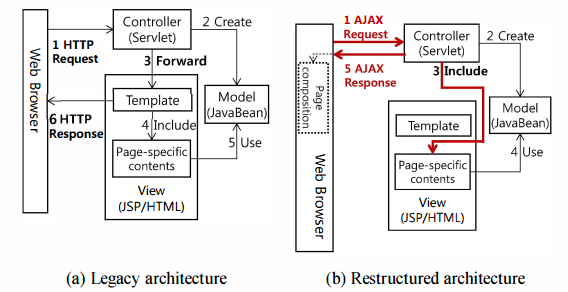


图3.客户端网页构成的体系结构转换

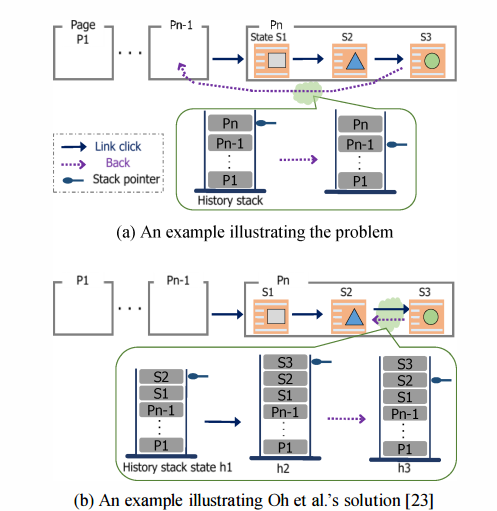


图4.向后导航问题和Oh等人的解决方案

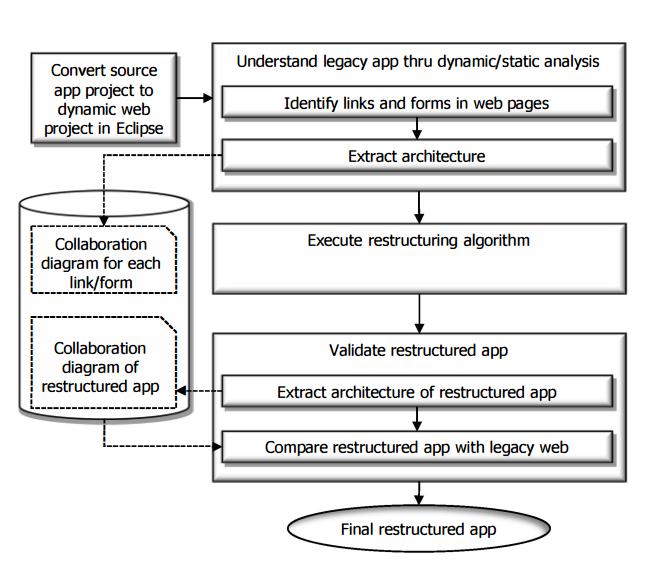


图5.整体重组过程

1. 重构算法

首先，解析TWA以收集一个整页请求列表，然后分析每个请求如果可能的话，通过使用修改为部分页面请求transformRequestO（见第1-15行）。 如果转换的请求将生成的SPA带入不一致的状态，即原始状态请求未被转换（见第5-7行）。 因此，输出应用程序可以是一个多页应用程序，其中每个页面作为单页面应用程序运行。 这种不一致的状态可以从MVC的角度出现三种情况架构。

第一种情况是转换的请求生成一个视图在TWA中不存在（请参见第20行）。 例如，假设我们有一整页请求r改变当前模板以及页面特定的内容。 如果r是转换为下载新的部分页面请求页面特定的内容，转换后的请求会产生一个不一致的观点。

模板不是一个静态页面，而是一个动态页面。不同的模板对象从模板实例化类，取决于系统/用户状态。 例如，在图中，模板对象在用户登录后被替换因为他们的菜单是不同的。 当用户请求需要一个新的模板实例，SPA也应该下载新的实例。 其中一种可能的设计是下载旧模板对象和新模板对象之间的差异。但是，这些差异可以分散在整个世界模板对象，所以实现并不容易。 因此，我们做出不转换原始整页请求的决定。

第二种情况是原始图像的转换请求导致不适当的情况：1）无效的控制器引入SPA，或者2）向不同的请求提出请求控制器超出了我们在TWA中的预期（见21-23行）。

首先，网页浏览器在HTML中执行客户端代码，级联样式表（CSS）和JavaScript。其中，JavaScript代码扮演着客户端控制器的角色。关于客户端代码的重要事项之一是当页面特定的内容被替换时，其副作用从旧的特定页面执行JavaScript代码内容仍然存在，不像HTML和CSS代码。在另一方面，整页请求会卸载所有的客户端代码旧页面。因此，需要分析以确定是否过时JavaScript代码让SPA的运行方式与TWA不同。改变可观察行为的具体场景的TWA如下：例如，假设一个页面特定的TWA中某个页面的一部分有JavaScript代码只要用户滚动到底部就下载图像页面（例如，通过AJAX请求中的window.onScrollO事件处理程序），以减少页面加载时间。如果用户移动到一个新的物理页面，这样的事件处理代码由于页面而自动移除卸下。但是，在重组的SPA中，代码被删除不会自动发生，因为进入一个新的状态不会导致整页刷新。这意味着一个无效的控制器存在于SPA（第21行）。

其次，SPA应该保留页面导航模型（包括向后/向前导航和刷新）输入TWA。例如，假设TWA的页面PI是重组为SPA页面p“，以便Ps与PI相同两页被刷新，新页面也应该是相同的。为了实现这一点，SPA的每个状态都有一个唯一的URL，例如TWA的每一页。这种状态的URL需要包含用于标识状态的信息（例如，请求URL获取其页面特定的内容）。因此，当一个页面是刷新，页面特定的内容可以一并获得该模板（见图8）。有一件重要的事情要注意当将整页请求重构为部分页面时请求使用AJAX：初始请求被重定向时（通过从服务器到HTTP的重定向请求web浏览器）到另一个组件c，重定向的URL（即，c）的请求URL，默认情况下不能通过温泉。另一方面，重定向的URL显示在带有TWA的Web浏览器的地址栏。因此，当我们尝试刷新SPA的页面，其中包括一个HTTP重定向请求作为页面特定的内容，SPA运行与其输入TWA不同。这意味着SPA发送向不适当的管制员提出要求（22-23行）。

第三种情况是转换请求可以生成的地方TWA中不存在的模型（参见第24行）。对于实例，原始请求的转换持续更新服务器状态可以允许生成SPA有无效数据。更具体的情况是如下：假设在TWA中，我们有一个页面不能使用相同数据提交的付款表单不止一次。但是，如果它的转换形式被允许两次提交相同的数据，转换不利地招致双重付款，这是一个例子双提交问题。