

SOCIAL MEDIA MASHUP FOR TURIST BOOKING SERVICES

Annamária Váradyová

School of Computer Science, Social Media and Web Technology, Linnaeus University,
Sweden

av222xc@student.lnu.se

Introduction

The key driver of the internet is information, we have never had as much information available or as readily available as today, just a few clicks. The web mashup (web application hybrid) is a web application that takes information from one or more sources and presents it in a new way, with a new use. The web is increasingly open and social. For this reason, many websites have open APIs that allow developers to access their core information. The architecture of a mashup is composed of the User Interaction Layer, Web Services and Data. [1]

The User Interaction Layer is the user interface of the mashup, we use technologies such as HTML, CSS, JavaScript and so on. The Web Services are accessing product functionality with API services. Tools: XMLHttpRequest and SOAP. We use technology JSON and XML for data handling like sending, storing and receiving data. A prime example of this is Google Maps, which is a trendy interface for use in mashups. Google allows developers to access their maps through APIs. The developer can then combine these maps with other data streams to create something new and unique. New types of mashups are being developed every day. This creates many new combinations with new Web 2.0 applications. Popular ones include Mashups with eBay, Amazon, Flickr, Digg and Twitter. [1][2]

Problem definition: Approach to solving the given problem and tasks

This document includes the main idea, rationale and novelty of how you have developed the resulting MashUp, the overall descriptions of the new components, diagrams, APIs and the MashUp patterns we have used. We also provide an evaluation strategy to evaluate our mashup in a particular dimension. For implementing it, we select and use the web programming language ReactJS. The goal of this assignment is to create and put into use a new MashUp based on services that already exist. We are developing a MashUp that incorporates data from three various social media applications kinds. Additionally, it offers users interaction rather than just a layer for data presentation. In order to handle the same origin, we construct a backend system that combines data from two independent services. [2]

Methodology approach:

Before designing and implementing the concepts we are proposing, we must first describe some user scenarios that show our ideas for the implementations and their related needs.

Scenario

A travel planner is a web application allowing users to plan every hour spent on their dream vacation. The application offers a wide range of services, search for hotels, restaurants, attractions, healthcare, tourism, ski resorts, etc. You can also view the weather forecast at the given destination—further routes, and ways to get to unified services. The user can plan each day separately: what, when, how, with what, and where he will spend his time.

Using the web application:

1. The user registers.
2. He is looking for a destination where he wants to spend his vacation.
3. He chooses the period from when he wants to travel there.
4. Choose the accommodation where you will spend your vacation.
5. Plan your days. He chooses the exact time when he wants to visit, for example, an attraction and can also find a way to get there. He continues with this method until he has scheduled all the days.
6. Finally, save your itinerary, which you can edit if you want.
7. Before leaving the application, write a message about the upcoming date of your trip.
8. During your vacation, you can monitor and shorten individual services or make other changes if necessary.

Architecture of web app:

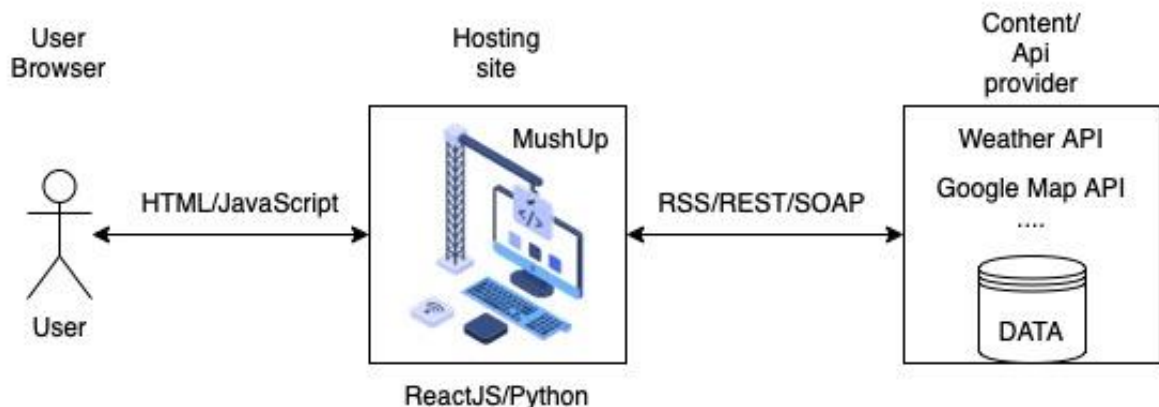


Figure 1 - Architecture of web app

Figure 1 shows the MashUp application architecture, which consists of 3 parts:[3]

- The user browser - The program is presented graphically and is used for user interaction within the consumer's web browser. The Mashups frequently use client-side logic to construct and assemble the mashup material.
- The MashUp hosting site - where the Mashup is hosted, the logic part of the web page
- The API providers - API providers are the content providers that are being mashed up. Providers often make their content available via web protocols such as RSS, REST and SOAP to facilitate data retrieval.

Outcomes/Analysis of results

The current scenario of the website:

The development of our website is at the beginning. For now, it only fulfils the primary function, searching for the place of our vacation, finding hotels, restaurants and attractions in the given location and saving them.

We would like to know if our application is available and if it is usable by users.

API's:

Google Map API

Rapid API – Travel advise. This API reproduce public data and features of TripAdvisor¹.

Evaluation strategy:

Our evaluation strategy is the practicality of the Travel planner web app. It is divided into two parts: user experience support and functional support.

- The user experience support - focuses on the usability of Travel planner from the standpoint of user interaction.
- Usability support - analyses how Travel planner supports user needs

Our strategy is based on the QUIM method. The Quality in Use Integrated Map (QUIM), which combines many elements, criteria, metrics, and data established in several human-computer interface and software engineering models, is presented for specifying and recognising quality in use components.[4][5]

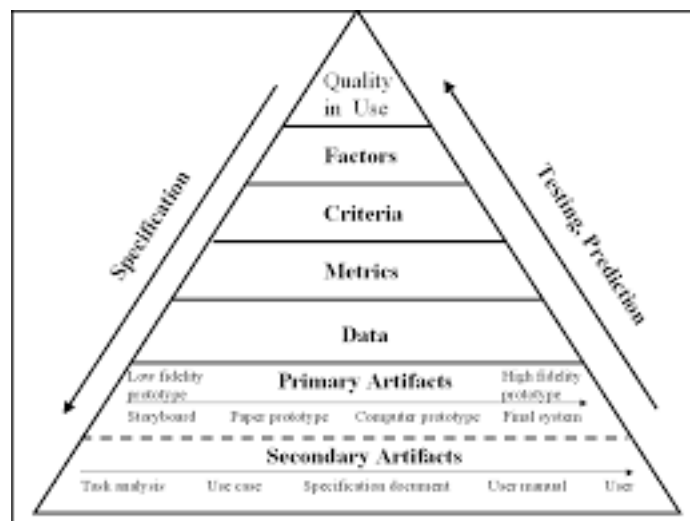


Figure 2 - QUIM model

Quality in Use:

Our goal is to provide as simple an application as possible for users. Therefore, we decided to focus on their opinion regarding the use and quality of the web application.

The developer and the end user observe the application from different points of view, so it is important to get information from the end user.

Factors:

As we mentioned above, our evaluation strategy also represents our factors.

- The user experience support
- Functional support

Criteria:

¹ <https://www.tripadvisor.com>

Criteria in the model also represent sub-factors. The difference is that they can be measured using a metric.

- The user experience support
 - clarity
 - consistency
 - minimal action
- Functional support
 - minimal action
 - completeness
 - minimal memory load

Clarity - A simple ability to perceive components. The necessary Mashup components should be readily apparent to the Travel planner without requiring much mental effort.

Consistency - The extent to which semantically comparable syntactic or visual components are expressed.

Minimal action - The connection between the elements, which build the mashup and the result to achieve as part of the mashup process.

Completeness - Changing a component can have unexpected consequences, or the change has an unexpected effect on the entire program.

Minimal memory load - Represents the amount of information that the user must remember to complete the task

Metrics:

In the QUi model, metrics represent a numerical value that summarizes the state of the given criteria.

We will evaluate our web application based on 5 criteria, which we described above in Criteria. To learn from the end users' opinions, we created 2 questions for each criterion that our respondents will answer.

- Clarity
 - Is the web application, Travel planner, easy to understand?
 - Is it correct to search using a travel planner?
- Consistency
 - Do the individual components of the web application, Travel planner work together correctly?
 - Are all of the parts that web application offers pertinent?
- Minimal action
 - Is it simple to utilise the web application?
 - Is it easy to recognize the individual procedures in the web application?
- Minimal memory load
 - Does the program clearly define the individual steps that the user must perform?
 - Is the information offered by the web application sufficient for its use?
- Completeness
 - Is the web application process straightforward and self-explanatory?
 - If there are any issues in the web application, are they evident?

Data:

We will collect data using a questionnaire in which there will be questions from Metrics. To evaluate individual questions, we need numerical values, and therefore individual answers will not be written, but the respondents will check the agree-disagree box. There will be 5 shelves,

where 1 represents agree and 5 represents disagree. Based on this data, we can make various analyzes and from the result, we will find out how the end users can work with the given application and whether they are satisfied with it and whether they would like to continue using the given application.

In the end, the survey is sent back to the developers, who will continue the development and improve the given requirements from the survey.

- Collecting
- Analysing
- Update the web app

Conclusions and Reflections

In the Introduction part, we clearly explained what MashUp is. It represents a web application that combines various data from various sources, mostly using APIs.

In the next part, Problem definition, we described what the given document should contain, and we analyzed individual parts in other sections. In the Methodical approach section, we presented our vision of the final web application. Our scenario is to create a web application that will be full-fledged and create a dream vacation for the user. The user can easily and simply plan every minute of their vacation with a few clicks. At the same time, he will be able to save and watch or edit it.

For the development of a web application, it is necessary to research whether our visions are realistic and whether the end user will welcome our proposal. In the section Results/Analysis of results, we described where our web application is located and we also brought an evaluation strategy. In the beginning, our web application only fulfils mainly the requirements, searching for hotels, restaurants and attractions in the selected location, and then we can save them in the calendar. In the course of development, we need some information from users. We built an evaluation strategy on this. It remains on the QUIM method, where we presented factors, criteria, metrics and data. The main idea of the strategy is the creation of a questionnaire in which the questions are created on the basis of the QUIM method, which we described in detail in this section. The questions are related to the user interface and functionality of the given web application. Subsequently, an analysis can be made from the survey, which will be used to continue the development of the given web application.

REFERENCES

- [1] Beemer, B., & Gregg, D. (2009). Mashups: a literature review and classification framework. *Future Internet*, 1(1), 59-87.
- [2] Liu, X., Hui, Y., Sun, W., & Liang, H. (2007, July). Towards service composition based on mashup. In *2007 IEEE Congress on Services (Services 2007)*
- [3] O'reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. *Communications & strategies*, (1), 17.
- [4] Seffah, A., Kececi, N., & Donyaee, M. (2001, December). QUIM: a framework for quantifying usability metrics in software quality models. In *Proceedings Second Asia-Pacific Conference on Quality Software*

- [5] Lin, H. X., Choong, Y. Y., & Salvendy, G. (1997). A proposed index of usability: a method for comparing the relative usability of different software systems. *Behaviour & information technology*, 16(4-5), 267-277.

HOSTING

<https://anna8295.github.io/travel-planner/>

SOURCE CODE

<https://github.com/Anna8295/travel-planner>