Challenge Outcome and Conclusion

Cesare Pautasso[1] and Florian Daniel[2]

[1] Faculty of Informatics, University of Lugano (USI), Switzerland
{name.surname}@usi.ch
[2] University of Trento, Via Sommarive 9, 38123 Povo (TN), Italy
daniel@disi.unitn.it

Abstract. In the following we report on the outcome of the ICWE 2015 Rapid Mashup Challenge (RMC), describe the voting system used, and draw some conclusions regarding the presented works.

Keywords: Mashups, Challenge, Benchmarking

1 Challenge Organization

We recall that every tool participating in the challenge was allocated 10 minutes for a short presentation with the goal to introduce the tool, illustrate its design and enumerate its most important features. Some participants also used the time to present the mashup to be built and discuss their choice of required Web APIs to be mashed up with others they could freely choose and how they were going to use their tool to assemble the mashup.

The demo part was also 10 minute long, during which the mashup was developed in front of the audience. The starting point for all demos was an empty workspace in which the components to be used in the mashup had been preregistered and pre-defined, but not yet assembled. Some authors chose to follow an iterative process, whereby the mashup was grown incrementally, piece by piece. Others also included a more general overview of the mashup tool capabilities, which was useful to demonstrate the expressive power of the tool, but did not necessarily help them build the most impressive mashup during the allocated time frame.

Each time a mashup was complete and the time for the demonstration had expired, the jury and audience had the opportunity to ask questions to the authors. This short interactive session had not been originally planned, but was very useful to provide the mashup authors with valuable feedback. During the same time, the challenge evaluation was collected through the ASQ system. The results were aggregated and the challenge ranking updated and shown to the audience and the tool authors.

2 The ASQ Voting System

The challenge evaluation phase was supported by the ASQ system [1]. ASQ (a permutation over Slides-Questions-Answers) allows anyone with a Web browser to follow a slideshow presentation and interact with the content by answering questions embedded in the slides. It was originally developed at the USI Faculty of Informatics to support in-classroom teaching activities by taking advantage of the fact that every student comes with his/her laptop to follow the lectures. Students not only can better read the content broadcast to their devices, but teachers can get real-time feedback about their level of understanding and thus adapt their pace and explanation depth during the lecture.

As such ASQ is a general tool and can be used also for any interactive presentation. In particular for the RMC, ASQ was extended with the following features:

- A special question type to gather ratings, over a 5-star scale, with the possibility to award also half stars.
- A count-down timer activated at the beginning of each demonstration to ensure every participant demonstrates his/her tool during the same amount of time.

The intention of introducing ASQ during the RMC was to broaden participation in the evaluation of the challenge participants from the jury to the whole audience (including the authors themselves, who did however not vote in their own turn). A secondary goal was to automate and increase the efficiency of the scoring process, where the answers are aggregated and the final ranking is recomputed after every participant is evaluated. Additionally, the slides showing the metadata about the current participants were interleaved with the questions to evaluate them. This helped to focus the jury's and audience's attention and build a shared awareness of the proceedings of the challenge and manage the time without introducing unnecessary delays.

3 Evaluation Criteria

In line with the call for participation of the RMC, every demonstration was evaluated according to four different criteria:

1. *Mashup Idea.* This focused on the functionality of the mashup to be assembled in under ten minutes. Also it took into account how the authors choose to combine the required APIs with others, if at all. The usefulness of the mashup also would come into play concerning this criteria.
2. *Mashup Complexity.* Given the strict time limit of 10 minutes, the complexity of the mashup is the challenging aspect. How complex can a mashup actually be when built in such a short time? This criteria was added also to measure the difficulty of building the envisioned mashup idea.
3. *Mashup Solution Elegance.* This criteria shifts the focus to the mashup implementation in the context of the specific mashup tool. The elegance, simplicity and understandability of the resulting mashup solution are all very important aspects that should not be underestimated, despite the emphasis we gave to the speed with which the solution has been assembled.

4. *Tool Power*. Based on the demonstration of the tool, seen in action for 10 minutes to build a specific mashup, the audience could also reflect on their impression of the tool's expressive power. Thus, this criteria does not reflect a complete analysis of the features of a given tool, but only what could be demonstrated in the limited time available.

4 Results

Table 1 summarizes the feedback obtained from the jury and the audience for each of the tools participating in the challenge in order of presentation.

Table 1: Feedback gathered from the jury/audience during the challenge

Tool	Mashup Idea	Mashup Complexity	Mashup Solution Elegance	Tool Power	Number of Votes
FlexMash	3.35	3.54	3.15	2.92	13
UI-Oriented Computing	3.07	2.57	3.18	2.36	14
SmartComposition	2.79	2.79	2.61	2.79	14
EFESTO	3.29	3.36	3.29	3.82	14
WebMakeup	2.61	2.21	2.64	2.96	14
WLS	3.07	2.90	2.70	3.10	15

Overall, the range of points collected by the tools is rather narrow, from 2.21 (the Mashup Complexity of WebMakeup) to 3.82 (the Tool Power of EFESTO). This shows that the audience – from a minimum of 13 to a maximum of 15 people provided feedback – provided a set of varied ratings, and that there is still room left for improvement in all criteria.

Concerning the Mashup Idea criterion, the tool ranked highest was FlexMash (3.35), which also scored highest (3.54) in the Mashup Complexity criteria. EFESTO, on the other hand, was ranked first according to both the Mashup Solution Elegance (3.29) and Tool Power (3.82) criteria.

Combining all criteria with equal weights led to the final ranking in Table 2, according to which EFESTO was awarded the first place in the ICWE 2015 Rapid Mashup Challenge.

Table 2: Ranking of the tools participating in the 2015 Rapid Mashup Challenge

Position	Tool	Total score
1	EFESTO [2]	13.75
2	FlexMash [3]	12.96
3	WLS [4]	11.77
4	UI-Oriented Computing [5]	11.18
5	SmartComposition [6]	10.96
6	WebMakeup [7]	10.48

5 Limitations

Given the wide variety of approaches to mashup tool design, both from research and industry, and the lack of standard or commonly accepted benchmarks to assess development tools, it remains difficult to give a fair comparison of mashup development tools. To provide an as representative picture as possible of the state of the art in mashup development, the RMC was intentionally left open concerning the type of tool admitted and challenged instead the participants with the rapidity of mashup assembly as the main constraint to compare the tools.

During the challenge, tools were demonstrated by their own authors, something that may invalidate any claim of usability or accessibility, especially by end-user programmers, usually associated with mashup tools. However, since every tool was used by the corresponding authors, the fairness of the comparison is not affected.

Concerning the use of the Web APIs, the second constraint of the challenge, Table 3 shows which APIs were used by each tool. The required APIs were announced one month in advance, giving the authors plenty of time to prepare. If one would want to stress the ability of tools to integrate heterogeneous Web APIs, components and data sources, this time could be reduced while increasing the number of required components in future editions of the challenge.

Table 3: Web APIs composed during the challenge by each mashup tool

Tool	score
FlexMash	NYT, Twitter
UI-Oriented Computing	NYT, Discover Magazine, Yandex Translate
SmartComposition	NYT RSS, AlchemyAPI, YouTube Search, TextTrack, Google Maps, Twitter, Wikipedia, Google Images
EFESTO	Song Kick, YouTube, Vimeo, Google Maps, Google Images
WebMakeup	NYT, NBC News, Google Search, Visual Economy
WLS	Twitter, Google Maps, GeoNames

6 Outlook

Concluding, we consider this first edition of the Rapid Mashup Challenge a success, from the point of view of both the quality of the presented mashup approaches (and authors) and from the number of participants overall to the event (about 30 people throughout the whole event). While on the one hand we have to register that comparing approaches for mashup development that are very different and diverse in their features is intricate and nontrivial, on the other hand, we also have to acknowledge that it is exactly this diversity and the distinguishing features that make the comparison (and the Challenge) interesting. So, the challenge for the future editions of the Challenge – and the process we wanted to start with this first edition of the Challenge – is to identify the right benchmarking approach for mashup tools, while staying open to all kinds of approaches the research community may come up with. This obviously represents a long-term objective, to be achieved over multiple iterations.

The next edition of the Rapid Mashup Challenge will take place at the 16th International Conference on Web Engineering (ICWE2016) next June 9th, 2016 in Lugano, Switzerland.

Acknowledgment We would like to thank the participants for their enthusiasm and the jury and audience for their active help with the evaluation of the presented approaches. We would also like to thank Vasileios Triglianos for his help and support with the ASQ tool.

References

1. Triglianos, V., Pautasso, C.: Interactive scalable lectures with ASQ. In: Proc. of the 14th International Conference on Web Engineering (ICWE 2014), Toulouse, France, Springer (July 2014) 515–518
2. Desolda, G., Ardito, C., Matera, M.: EFESTO: A platform for the End-User Development of Interactive. In: ICWE 2015 Rapid Mashup Challenge. Volume CCIS 591., Springer (2015) 62–80
3. Hirmer, P., Mitschang, B.: FlexMash Flexible Data Mashups Based on Pattern-based Model. In: ICWE 2015 Rapid Mashup Challenge. Volume CCIS 591., Springer (2015) 12–29
4. Gallidabino, A., Babazadeh, M., Pautasso, C.: Mashup Development with Web Liquid Streams. In: ICWE 2015 Rapid Mashup Challenge. Volume CCIS 591., Springer (2015) 97–116
5. Nouri, A., Daniel, F.: Interactive, Live Mashup Development through UI-Oriented Computing. In: ICWE 2015 Rapid Mashup Challenge. Volume CCIS 591., Springer (2015) 30–48
6. Krug, M., Wiedemann, F., Gaedke, M.: SmartComposition: Extending Web Applications to Multi-Screen Mashups. In: ICWE 2015 Rapid Mashup Challenge. Volume CCIS 591., Springer (2015) 49–61
7. Díaz, O., Aldalur, I.A., Arellano, C., Medina, H., Firmenich, S.: Web Mashups with WebMakeup. In: ICWE 2015 Rapid Mashup Challenge. Volume CCIS 591., Springer (2015) 81–96