

Initial Plan

Olympic Diving Results Information System Powered by Olympic Data Feed

Author: Matthew Aish
Supervisor: Dr. J. Shao
Moderator: Prof. D. Marshall
Module Code: CM0343
Credits: 40

Project Description

Beginning in July 2011, I started working for LOCOG (London Organising Committee of the Olympic Games) where my role was a Test Analyst and Deputy Results Manager for Diving. The bulk of this role involved testing the various components of the Diving results systems, this included:

- The LOCOG games-time website - providing live results to the world
- CIS (Commentator Information System) -providing commentators and journalists with touch-screen technology delivering results in real time
- Info -a web application acting as the information hub of the games.

These systems are powered by ODF (Olympic Data Feed), an XML feed that is sent from the venues back to in house systems and then forwarded to the applications as well as press and broadcast agencies around the world. The feed consists of point in time messages, which are sent prior to and after competition (this includes participant information, start lists and medallists), and real time messages during competition, which contain all results and current standing information as the competition progresses.

After my experience testing and providing input as these applications were developed, I am going to implement my own results system for Diving using sets of ODF messages collected from testing. The original feed is no longer active with the conclusion of the games, so I will be using the raw xml files. I plan to implement a system that improves on and adds new features to those that already exist:

- The previous systems simply allowed users to absorb information. I want to incorporate greater user interaction through the use of prediction. For example, users could predict scores as the competition progresses, which could lead to a game-like prediction league.
- I want to make greater use of the information that is available in the xml messages. For example, detailed scoring information was only available immediately after a dive rather than throughout the competition; and some systems, such as CIS, would only show a single days events and would lose all data at the end of the day.
- In terms of the implementation, there will be major differences. Although I extensively tested the existing systems this did involve any interaction with the source-code or architecture and mainly involved integration and user testing.

Therefore my approach to delivering my aims and objectives (specified below) is likely to be very different from what was conducted previously.

Aims and Deliverables

- Implement a dummy version of the Diving Olympic Data Feed so that messages can be sent and received, and then processed.
- Parse ODF messages and store the data contained in a database to allow meaningful use.
- Provide an interface for users to view results information, including:
 - Schedule information
 - Participant information (entries, team composition, biographical data etc.)
 - Start lists and results
 - Medal winners and final rankings
- Provide results information (previous diver scores, current standings etc.) in real-time as ODF messages are received during competition.
- Allow users to interact with the system during competition by predicting scores and medallists.

Expected inclusions for the interim report:

- Background research
- Detailed analysis and approach to development
- Design and implementation of the dummy feed
- Design and implementation of the database
- Implementation of message processing (I aim to have this completed but this may overrun)

Expected inclusions for the final report:

- Detailed designs of the user interface and user interaction
- Complete implementation meeting deliverables
- Analysis of implementation, design and approach
- Conclusions and learnings

Work Plan

In terms of the work plan, I have split the development of the project into 4 main areas:

- Implementing the dummy ODF feed
- Processing messages and inserting the data into a database
- Implementing a user interface
- Providing user interaction by predicting outcomes

I think this is the most sensible approach to take to implementation, as each area is largely dependent on the success of its predecessor. For example, until I have implemented the sending of XML messages, it is difficult to know exactly how to process them and insert the data into a database. Until the database is complete, it will be difficult to know how to build the user interface. And without a baseline user interface, it will be impossible to incorporate the user interaction features. I therefore see these four areas as the places throughout the project timeline where there is the greatest likelihood that the project direction may change (for example, needing to change my approach to implementation, new ideas for the use of the application etc.). By completing these core elements before beginning work on the next, I believe I will minimise the risk that the direction will change drastically, resulting in completing redundant work, and ultimately minimise risk of project failure.

It is still very important to acknowledge that difficulties, problems and changes are likely to occur during the project. Because of this I have planned slack and tried to stagger tasks as much as possible to allow time for these cases.

After the Christmas and examination periods, I intend to focus on development of the user interface and user interaction features. Currently it is difficult to specify exactly what this will entail as the project may have gone through many changes by this point. I have decided that the best way to specify these tasks at the moment is in cycles- i.e. a baseline, followed by a second version with complete functionality, and a final version to fix any errors or small features that may be missing. In the New Year, I should know exactly what I am going to implement so this part of the plan will be rethought and planned in greater detail at that time.

It is also worth noting that I am only planning to start the reports 3 weeks before the submission deadline. In reality, I plan on keeping logs and designs as I work, so this short period of time will mainly comprise of formalising these notes to complete the reports, rather than giving myself a lot of work to complete in a small amount of time.

The next pages show a Gantt chart of the entire project timeline, including task breakdowns with estimations of their durations.

ID	Task Name	Duration	Start	Finish	15 Oct '12							22 Oct '12							29 Oct '12
					S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	Initial Reasearch and Design	8 days	Mon 15/10/12	Wed 24/10/12															
2	Background reading and research	6 days	Mon 15/10/12	Sun 21/10/12															
3	Basic Design	3 days	Mon 22/10/12	Wed 24/10/12															
4	Data Feed	13 days	Thu 25/10/12	Mon 12/11/12															
5	Design "data feed" implementation	4 days	Thu 25/10/12	Tue 30/10/12															
6	Implement "data feed"	3 days	Wed 31/10/12	Fri 02/11/12															
7	Finalise data to be captured from feed	2 days	Wed 31/10/12	Thu 01/11/12															
8	Implement XML parsing	6 days	Mon 05/11/12	Mon 12/11/12															
9	Database and XML Processing	23 days	Wed 07/11/12	Sun 09/12/12															
10	Design database	4 days	Wed 07/11/12	Mon 12/11/12															
11	Implement database	5 days	Tue 13/11/12	Sun 18/11/12															
12	Create DB cleaning functions	2 days	Tue 13/11/12	Wed 14/11/12															
13	Implement config messages processing	6 days	Mon 19/11/12	Sun 25/11/12															
14	Implement PIT results processing	11 days	Mon 26/11/12	Sun 09/12/12															
15	Implement RT results processing	11 days	Mon 26/11/12	Sun 09/12/12															
16	Interim Report	19 days	Mon 19/11/12	Fri 14/12/12															
17	Interim Report first draft	16 days	Mon 19/11/12	Sun 09/12/12															
18	Interim Report revisions	4 days	Mon 10/12/12	Thu 13/12/12															
19	Interim Report Submission	0 days	Fri 14/12/12	Fri 14/12/12															
20	User Interface	25 days	Mon 28/01/13	Fri 01/03/13															
21	Design UI	6 days	Mon 28/01/13	Mon 04/02/13															
22	Implement static UI pages	5 days	Tue 05/02/13	Mon 11/02/13															
23	Implement basic results pages for UI	13 days	Wed 13/02/13	Fri 01/03/13															
24	Second version of results for UI	13 days	Wed 13/02/13	Fri 01/03/13															
25	User Prediction	22 days	Mon 04/03/13	Tue 02/04/13															
26	Design user predication	5 days	Mon 04/03/13	Fri 08/03/13															
27	Implement user predication in UI	7 days	Mon 11/03/13	Tue 19/03/13															
28	Further develop user predication	10 days	Wed 20/03/13	Tue 02/04/13															
29	Final Report	24 days	Mon 01/04/13	Fri 03/05/13															
30	Final report first draft	17 days	Mon 01/04/13	Tue 23/04/13															
31	Final report revisions	7 days	Wed 24/04/13	Thu 02/05/13															
32	Final Report Submission	0 days	Fri 03/05/13	Fri 03/05/13															







