Crossroads
The ACM Student Magazine

TigerEvents: An Online Event Calendar System for Students by Students

By ***Chris Jordan***, ***Oliver Baltzer*** and ***Sean Smith***

Introduction

Over the past couple of years, members of the Dalhousie Student Chapter of the ACM, Dal-ACM, have been active in developing and maintaining open source software used by students. Consequently, Dal-ACM has become a well known student society at Dalhousie University for providing computer solutions to improve student life. Developing software for students is a good way for members of any ACM Student Chapter to get involved in their respective campus while gaining practical software development and project management experience. TigerEvents is one such software project organized by Dal-ACM members; it helps students find out about events that they would be interested in on campus.

University campuses are a jumble of activity. Nowhere else in our society do we find people from every corner of the world studying fields of every discipline. Universities are places where people broaden their horizons, and it is unfortunate that most students spend only four or five years in this environment. Many campuses have events running throughout the year, put on by different faculties, departments, and student groups. Typically, there is so much activity that it is difficult to communicate what events are happening on campus. Therefore, students oftentimes miss out on events that they would otherwise be interested in.

Dal-ACM saw that this problem could be addressed by a carefully designed software solution that provides students with a novel means of advertising events and communicating announcements. In the summer of 2005, members of Dal-ACM developed TigerEvents [9], an open source event calendar system. This Web-based application allows students, faculty members, and alumni to post events and make announcements. The postings are presented in an online newspaper format and organized by date, category, and the group that posted them. TigerEvents gives students, members of the community surrounding Dalhousie University, and visitors a consolidated view of campus events.

Although there are alternative methods for advertising events and making announcements, many students have found these methods to have drawbacks preventing them from being an effective means of spreading information over a whole campus. These methods include:

- Email:

  Mass e-mails are a quick way to spread information with little effort. These types of e-mails are, however, often treated as spam by the bulk of the student body. Attempts to save time by aggregating several topics into one e-mail are only superficially more palatable and are self-defeating, as students will be even less likely to read such a missive.

- Posters:

  Bulletin board posters do have some measure of success, mostly due to the visual elements being able to catch a reader's eye. However, they sometimes suffer from poor coverage if they are not placed along popular travel paths. An additional concern is the time required by an individual to canvas a large area. Also problematic is the average poster's life span, as they are prone to damage from passersby, lack of visibility when other posters are put over them, and removal at regular intervals.

- Word of Mouth:

  Word of mouth is usually quite effective. The premise is that a good event will generate buzz among a group of people, who then tell people they know about the event, and so on. The major drawback with word of mouth advertising is that it often takes longer to disseminate information within a large group. If an event is reoccurring, then this drawback is mitigated; however, many events only occur once.

- Websites:

  Websites are easily accessible to anyone with a computer connected to the Internet. These days, this level of connectivity is almost a certainty among students. Additionally, public computers allow access from a wide variety of locations on campus, making checking for events that much easier. Visual elements are easily provided to visitors of a Website. The drawbacks to this approach are still accessibility and time contraints. If a student does not know about a site that advertises particular events, they will still miss out on them. Furthermore, visiting all student society web pages (there are roughly 300 student societies at Dalhousie University) to search for events is an infeasible process.

This paper presents the inception of the TigerEvents project and our hard-earned experience in both project management and teamwork during its development and deployment at Dalhousie University. The aim of the paper is to give other ACM student chapters and student groups a road map on how to start an open source project on their campus for the benefit of other students. These projects can be new applications or can build off of existing ones, such as TigerEvents. We will cover parts of the project's development process that went well, discuss parts that were challenging, and finally look at where we are hoping to go with our future activities.

Getting Started

ACM student chapters are student organized and are integral parts of a University's campus. As such, they are confronted with the same issues that other student groups or individual students are facing. Many of those issues, however, can be overcome with software systems to, for example, aid the management of information or simplify reoccurring processes. Hence, due to its inherently broad knowledge of computer science skills, an ACM student chapter can be the perfect environment to develop software "by students, for students." Students are encouraged to participate in such research without fear of censure if they fail in such extra-curricular undertakings. Thus, they have the freedom to gain valuable experiences in software development and project management without any harm to reputation or academic standing if the project fails.

Typically, there is not a lot of money available for student organizations and societies that can be spent on software development. Thus, any project has to be able to operate with the resources that are readily available. Additionally, student groups are generally very dynamic, and membership is constantly fluctuating. Hence, they cannot provide any guarantees about the availability and the continuous maintenance of the software they provide. To increase the chances of a project's survival beyond its initial implementation and the original developers' graduation, a free software and open source approach is advisable. The very nature of open source software is that it can be used, modified, and redistributed without major restrictions and charges. This open nature grants access to its source code for everyone to make modifications and potential improvements to the software. For a student group, it also means that they can build their own software on top of existing open source projects and customize them to the needs of their campus.

Sometimes, problems or issues that can be resolved with the aid of computer software are hard to identify. The body of university students is usually very diverse, and a group of computer scientists may not be able to immediately identify issues that are of greater concern for all students. A good place to potentially get a lot of support and general information about students' needs is the student union or the highest governing student organization on campus. Often, those larger organizations themselves have processes that could be simplified through automation. Even though students may not be affected directly by those processes, they will benefit from the increased efficiency of their governing organization. For example, Dal-ACM started working with the Dalhousie Student Union in 2004 and developed the used books exchange website TigerBooks [8], to help students to find and sell used books on campus.

Working with a student union has the additional benefit of helping to promote the software and to reach out to other students and societies. In working with the Dalhousie Student Union (DSU), it has been Dal-ACM's experience that promotion of our software to students is entirely managed by the DSU, allowing Dal-ACM to focus on development, improvement, and maintenance of the software.

Gear that makes life easier

For open source development and teamwork to be successful, an appropriate infrastructure needs to be in place. This includes project management, issue tracking, software configuration management, as well as communication channels between the developers themselves and the users. There are many sites [3, 4, 5] available on the Web that provide complete services for open source development, including project management, software configuration management, issue tracking, mailing lists, and website hosting. Though it is not required, it is suggested to use one of those managed services to significantly lower the administrative overhead and computational resource requirements of a project.

At Dal-ACM, we decided to manage most infrastructure services ourselves, because we already had sufficient computing and administrative resources available. As a software configuration management and code repository, we deployed Subversion [6], a centralized revision control system. It contains all our development source code and allows for easy tracking of changes to it. Additionally, it provides management of different versions of the source code in parallel, such that concurrent development on the same code base is simplified.

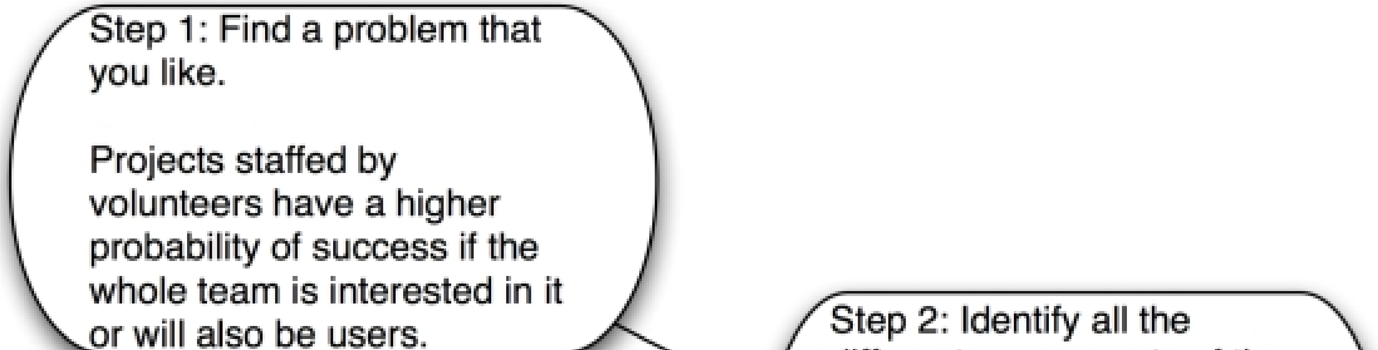
Depending on the size of the project and the number of people involved with it, a project management system may not be necessary, but if used properly, it is a valuable asset. Project management systems provide various mechanisms to organize tasks throughout the project and document the progress that is made. The typical components of a project management system are issue tracking and release management. Dal-ACM is using Trac [1] for this purpose. Trac is an open source project management system that integrates well with Subversion. It provides a fully-featured ticket system to manage tasks and issue. Additionally, it integrates a Wiki-style documentation system, a Subversion repository browser, and a milestone concept for release management.

An issue tracking or work ticket system is also an essential part of any project. In our case, as the developers were not always capable of meeting face to face, it was doubly essential to implement an issue tracking system in order for everyone to know what the other members were working on. This prevented individuals from duplicating work and, thus, saved time. Additionally, this helps with the resolution of software errors, because when a problem is discovered with the project, it can immediately be put into the issue tracker, instead of trusting the discoverer to bring it up at the next meeting.

Having these tools requires the use of a server. Fortunately for most projects, a reasonably standard desktop machine suffices. Until recently, Dal-ACM used a Pentium 4 desktop as a development server and a Pentium 2 desktop as a production unit. Given that we were working with limited resources, we also used Linux and FreeBSD on the development and production servers, respectively. This hardware should be fairly inexpensive to obtain. One important consideration is that the servers should take advantage of the high-capacity, low-latency links provided by the university network. Although some universities have implemented restrictions on bandwidth or other forms of traffic shaping, most still have fairly liberal policies, especially for machines that are meant for research or have a legitimate support role. The home ISP of a developer may have much more restrictive policies for incoming and outgoing connection rates and bandwidth usage.

Now the fun begins…

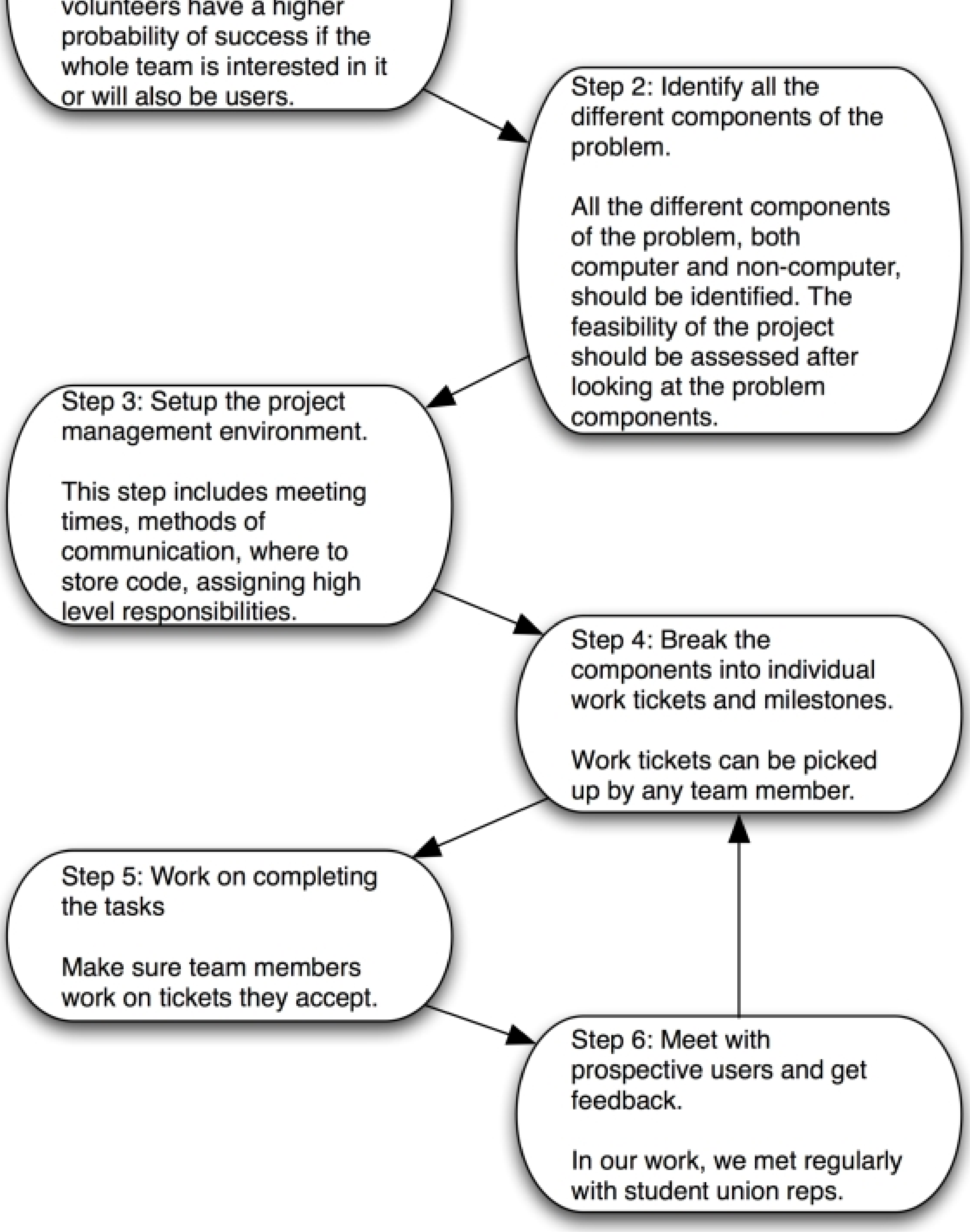
One thing you should not be afraid of is throwing out previously done work. The first prototype of TigerEvents was actually written in PHP, but ended up being discarded. Although we would prefer to call the process of creating that first version "rapid prototyping," the reason that we abandoned it is that PHP does not effectively separate the middle-tier logic from the presentation. As PHP is embedded in HTML, it naturally mixes the two, and this mixture leads to messy code. However, the experience of designing the first version was quite helpful. It is difficult to know what all the requirements are for the project without developing at least some sort of working prototype.



Step 1: Find a problem that you like.

Projects staffed by volunteers have a higher probability of success if the whole team is interested in it or will also be users.

Step 2: Identify all the



volunteers have a higher probability of success if the whole team is interested in it or will also be users.

Step 2: Identify all the different components of the problem.

All the different components of the problem, both computer and non-computer, should be identified. The feasibility of the project should be assessed after looking at the problem components.

Step 3: Setup the project management environment.

This step includes meeting times, methods of communication, where to store code, assigning high level responsibilities.

Step 4: Break the components into individual work tickets and milestones.

Work tickets can be picked up by any team member.

Step 5: Work on completing the tasks

Make sure team members work on tickets they accept.

Step 6: Meet with prospective users and get feedback.

In our work, we met regularly with student union reps.

Figure 1: The high level steps involved in running a successful software project.

To help with development, you should routinely meet with representatives from your Student

Union. It is important that you try to get as much input from students as possible, because they are ultimately going to be your main users. That being said, not everything they are going to ask for will be sensible. You have to find a balance between what they want and what you think is a sensible solution. Not getting user input puts your project in danger of not being accepted. On the other hand, listening too deeply to feedback can lead to you running in circles. Generally, Dal-ACM tries to meet with student representatives every couple of weeks to review new implementations and determine what new features should be added. Frequent short meetings are far better than infrequent long ones. It will make your team more responsive to student needs.

Fast Application Development with Ruby on Rails

Our second attempt at developing TigerEvents used the Ruby on Rails (RoR) framework [7]. Even though web programming has been possible in Ruby since its initial release, it was only based on the rather rudimentary Common Gateway Interface (CGI), which defines a simple communication interface between a web-server and a web-application. Rails is a new framework, released in July 2004, for Ruby and offers a means to quickly develop advanced Web applications in a maintainable fashion. It achieves this by applying a variety of software design and programming techniques, such as the Model View Controller (MVC) [2] design pattern and meta-programming.

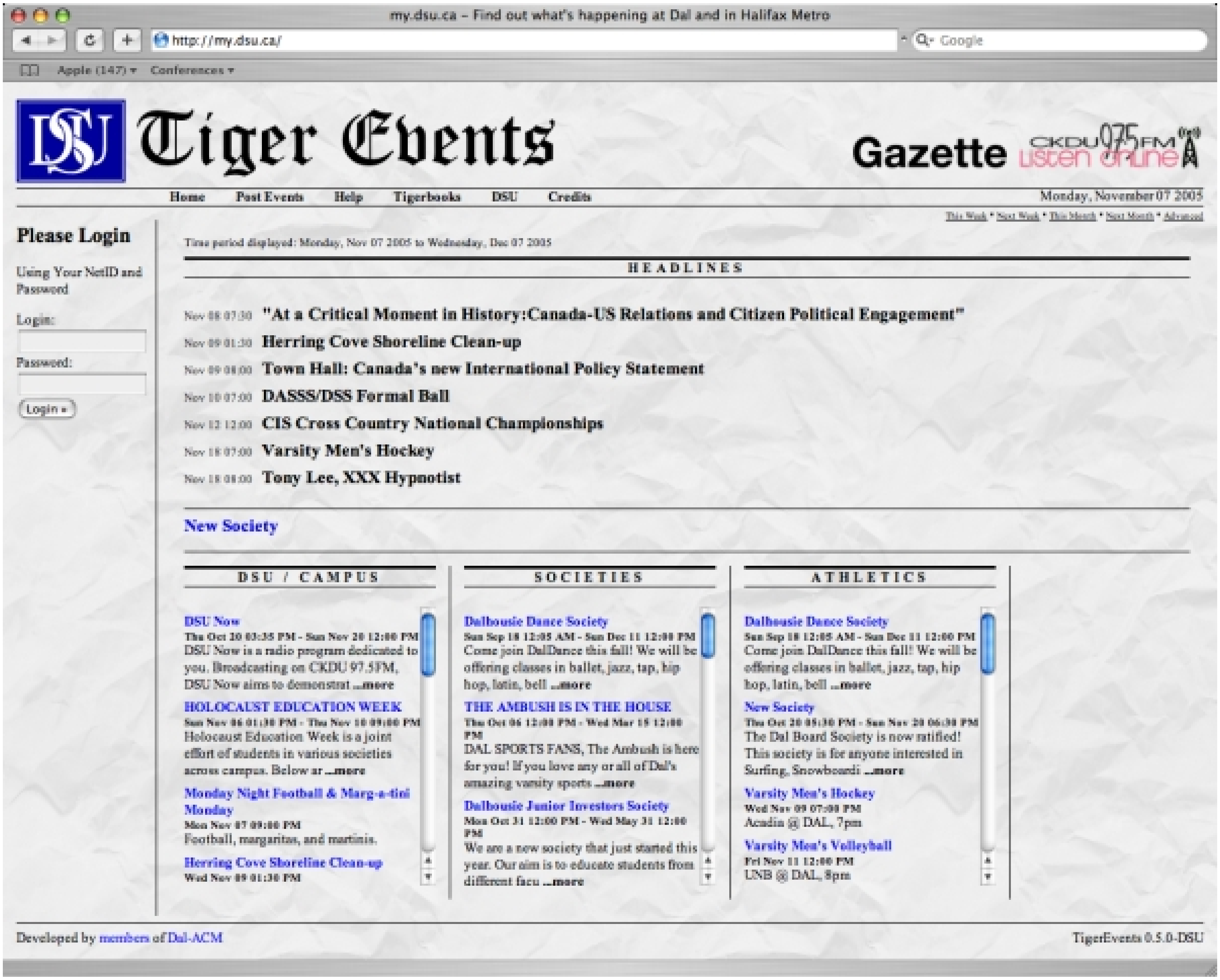
Using meta-programming, the RoR is able to dynamically map database properties, such as table or column names, as symbols into the application. This mechanism allows for a significant reduction in code for data access and manipulation. In fact, RoR makes it possible to insert, modify, and access records in a database without having to write a single SQL statement. This feature of RoR makes it attractive for advanced Web development because having to generate less code translates into faster application development. For example, the beta release for TigerEvents was created in just over a month by three students working part time.

Additionally, RoR follows a strict MVC methodology that separates the presentation of data from the application logic and the data access logic. Whereas meta-programming reduces the amount of code written for the data access logic, the Ruby standard library and various extensions provided by RoR reduces the amount of code for the application logic. The view logic or presentation of the data is generally implemented by using some templates that embed Ruby code for minimal control flow and access to a program´s internal objects. All components of the MVC model interact seamlessly through RoR such that they are implementation independent of each other, and there is no need for additional "glue" code.

All of these properties of RoR minimize the amount of code that has to be written in order to make a functional advanced Web application with a database back-end. By hiding the SQL logic, separating the component implementations, and providing powerful standard libraries, RoR helps construct more maintainable code. Proper database design, however, is fundamental to exploiting all of the features of RoR. Bad database design will result in unmaintainable code or database changes that will be costly in terms of the code modifications that have to be made to accommodate them. It is well worth the additional effort to design a database that will meet the needs of your system.

TigerEvents

TigerEvents is an online community-oriented event calendar system designed by members of Dal-ACM to meet the need for greater communication on the Dalhousie University campus. It provides students with an interactive Website where they can view events that are coming up and post events of their own. There are essentially three functions available to TigerEvents users: view upcoming events and announcements, post upcoming events and announcements for their respective group, and administrate their groups.



Home Post Events Help Tigerbooks DSU Credits

Monday, November 07 2005

This Week * Next Week * This Month * Next Month * Advanced

Please Login

Using Your NetID and Password

Login:

Password:

Login »

Time period displayed: Monday, Nov 07 2005 to Wednesday, Dec 07 2005

HEADLINES

Nov 08 07:30 **"At a Critical Moment in History:Canada-US Relations and Citizen Political Engagement"**
Nov 09 01:30 **Herring Cove Shoreline Clean-up**
Nov 09 08:00 **Town Hall: Canada's new International Policy Statement**
Nov 10 07:00 **DASSS/DSS Formal Ball**
Nov 12 12:00 **CIS Cross Country National Championships**
Nov 18 07:00 **Varsity Men's Hockey**
Nov 18 08:00 **Tony Lee, XXX Hypnotist**

New Society

DSU / CAMPUS

DSU Now
Thu Oct 20 03:35 PM - Sun Nov 20 12:00 PM
DSU Now is a radio program dedicated to you. Broadcasting on CKDU 97.5FM, DSU Now aims to demonstrat ...more

HOLOCAUST EDUCATION WEEK
Sun Nov 06 01:30 PM - Thu Nov 10 09:00 PM
Holocaust Education Week is a joint effort of students in various societies across campus. Below ar ...more

Monday Night Football & Marg-a-tini Monday
Mon Nov 07 09:00 PM
Football, margaritas, and martinis.

Herring Cove Shoreline Clean-up
Wed Nov 09 01:30 PM

SOCIETIES

Dalhousie Dance Society
Sun Sep 18 12:05 AM - Sun Dec 11 12:00 PM
Come join DalDance this fall! We will be offering classes in ballet, jazz, tap, hip hop, latin, bell ...more

THE AMBUSH IS IN THE HOUSE
Thu Oct 06 12:00 PM - Wed Mar 15 12:00 PM
DAL SPORTS FANS, The Ambush is here for you! If you love any or all of Dal's amazing varsity sports ...more

Dalhousie Junior Investors Society
Mon Oct 31 12:00 PM - Wed May 31 12:00 PM
We are a new society that just started this year. Our aim is to educate students from different facu ...more

ATHLETICS

Dalhousie Dance Society
Sun Sep 18 12:05 AM - Sun Dec 11 12:00 PM
Come join DalDance this fall! We will be offering classes in ballet, jazz, tap, hip hop, latin, bell ...more

New Society
Thu Oct 20 05:30 PM - Sun Nov 20 06:30 PM
The Dal Board Society is now ratified! This society is for anyone interested in Surfing, Snowboardi ...more

Varsity Men's Hockey
Wed Nov 09 07:00 PM
Acadia @ DAL, 7pm

Varsity Men's Volleyball
Fri Nov 11 12:00 PM
UNB @ DAL, 8pm

Developed by members of Dal-ACM

TigerEvents 0.5.0-DSU

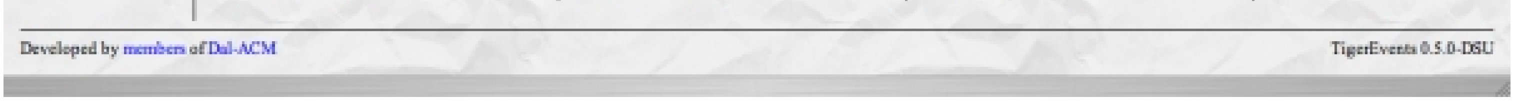


Figure 2: The TigerEvents main page displays upcoming events to users.

For viewing events, TigerEvents is different from other traditional calendar programs. Instead of displaying a grid of days for a particular month, view boxes, which we refer to as Selectors, display events with particular properties. For example, in our initial production release at Dalhousie, we have four Selectors: one for Student Union events, one for Student Society events, one for Athletic events, and one for Faculty events. The site administrators, who are also Student Union representatives, are able to select different events to be in the Headline Selector. We believe this approach to displaying events is better than the approach used on traditional day grids, as it helps avoid the "date clumping" problem, where multiple postings for a particular date can lead to a cluttered display. Additionally, we felt that it would be more beneficial to users to look for events by particular properties, not just time.

Conclusion

In less than two months, three students in the Dalhousie Student Chapter of the ACM (Dal-ACM) were able to develop the Beta version of TigerEvents. After a two week Beta, TigerEvents was deployed at Dalhousie University as another service of the Dalhousie Student Union (http://my.dsu.ca). Part of the reason for the fast development of TigerEvents can be attributed to the Ruby on Rails framework (RoR). RoR is a rich framework that provides many tools for advanced Web development and permits rapid prototyping and eases code writing. The use of a Subversion code repository and Trac project management system were also fundamental in allowing project members to work efficiently together.

In organizing projects such as TigerEvents for a university campus, it is important to be in regular contact with the Student Union / Association. Not only will they be a valuable source of feedback, but it will also help ensure that the project will be accepted by students. Having a good relationship with the Student Union is a crucial part of a project's success.

TigerEvents is very much an active Dal-ACM project. We are currently looking to incorporate syndication using iCalendar. It is our intent for users to be able to upload and download iCalendar data, which is used in popular calendar software such as Microsoft Outlook and Mozilla Calendar. We also are planning to develop new layouts for TigerEvents. Currently there is a single newspaper style layout for TigerEvents.

An additional feature that we should have incorporated into TigerEvents by the time this article is published are customized Selectors. In our initial production version of TigerEvents, 0.5, there were set Selectors, boxes that display events and announcements with given properties, for the Student Union, Student Society, Athletics, and Faculty. In our next release, we intend on providing users with the capabilities to customize what Selectors they have.

One of the errors that we committed while working on TigerEvents was not developing any standardized testing. Unfortunately, we overlooked the importance of having solid testing. Not having a test suite means that testing new features and changes takes much longer as all of it has to be performed manually. We are planning to create standardized testing for TigerEvents in the near future.

On a side note, we are collecting all the calendar data that is being posted on TigerEvents. Having this data will allow us to conduct more realistic and rigorous usability studies to look at what really is the most effective means for displaying calendar data across an organization. The results from this study will allow for better layouts to be developed.

We have shown here that it is possible for a student group like an ACM student chapter to develop Computer Science solutions for problems on their university or college campus. TigerEvents is one such system developed by the Dal-ACM. We also feel that it is important that these software projects be open source as well. Being open source allows a project to be shared by other student chapters at other universities and promotes the concept of creating software "by students, for students." We hope that this idea spreads to other universities and colleges, and perhaps some day in the future, there will be code exchanges and joint projects between different ACM student chapters. We welcome all student chapters to download TigerEvents from SourceForge, [9] as well as to make development contributions.

Acknowledgment

We would like to thank James Richard, a member of the Dalhousie Student Chapter of the ACM (Dal-ACM), for his user documentation and testing contributions. Also, we would like to thank Philip Duguay, the Dalhousie Student Union Vice President Internal, for his support on TigerEvents and other Dal-ACM projects. Chris Jordan is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Chris Jordan is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1

Edgewall Software: Trac. URL: **http://www.edgewall.com/trac/**

2

Reenskaug, T. *THING-MODEL-VIEW-EDITOR: an Example from a Planning System*

Xerox PARC Technical Note, May 1979.

3
Rubyforge. URL: **http://www.rubyforge.org/**

4
Savannah. URL: **http://savannah.gnu.org/**

5
SourceForge. URL: **http://sourceforge.net/**

6
Subversion. URL: **http://subversion.tigris.org/**

7
Thomas, D. and Hansson, D.H. *Agile Web Development with Rails.* The Pragmatic Programmers LLC, 2005.

8
TigerBooks. URL: **http://www.dal-acm.ca/Projects/TigerBooks**

9
TigerEvents. URL: **http://www.tigeronrails.com/tigerevents**

Biography

Chris Jordan (**cjordan@cs.dal.ca**) is a PhD candidate in the Faculty of Computer Science at Dalhousie University. His research interests are in Information Retrieval, Natural Language Processing, and Web Mining. He is an active member of his local ACM student Chapter, Dal-ACM, and enjoys working on open source software projects. Website: **http://www.chrisjordan.ca**

Oliver Baltzer (**obaltzer@cs.dal.ca**), PhD candidate at the Faculty of Computer Science of Dalhousie University, is a passionate geek who enjoys writing open source software and being involved with things that matter.

Sean Smith (**ssmith@cs.dal.ca**) is an undergraduate student at Dalhousie University. He is a strong believer in being active in the community, being an active member of the Computer Science Society and The Dalhousie Student Chapter of the ACM.