

An Online System for Visualizing UOIT Class Schedules



UNDERGRADUATE HONOURS THESIS
FACULTY OF SCIENCE (COMPUTING SCIENCE)
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Bradley Chicoine

SUPERVISOR:
Jeremy Bradbury
Christopher Collins
Faisal Qureshi

April 25, 2010

Abstract

The objective of this thesis is to design a system that will enable users to visualize aspects of a schedule so that the information can be interpreted in a meaningful way. The concept of visualizing data has allowed many possible forms such as graphs, time tables, maps, etc. In order to make the results of this thesis more accessible it was decided early on to switch to a web-based implementation. This way a wide audience of faculty or students of the school could have access this information and allow for the project to have a larger scope.

Implementations for displaying data include tables, a week calendar view, a week time table view with classes sized to represent time, and coloured bar graphics to represent capacity and time based on spatial location. In addition to the method of displaying schedule information the data to display has been focused on a few different areas of: courses, rooms, and capacity. Displaying lists of courses as well as being able to connect courses to professors resulted in the calendar view. The need for room schedules resulted in the time table to better display the information and help with the availability visualization at the same time. However the availability of several rooms or several professors was not capable with the calendar or time table view, so this resulted in more abstract methods of attempting to display availability data. Availability is displayed using generated graphics of coloured bars representing classes with their colour representing capacity.

Another requirement besides the visualization part of each piece of the thesis was the accessibility. Being a web-based program the project to be flexible such as allowing different parameters to customize results, as well as being easy for the average user to navigate. Methods such as connecting different colours for each visualization, breadcrumbs, tooltips, clickable areas to display more information and easy menu selections were included in the thesis webpage. This project has iterated over several different designs through web pages and visualization to attempt to find the most flexible method of presenting the collected information.

Contents

1	Introduction	1
1.1	Problem Statement and Motivation	1
1.2	Goals	2
1.3	Thesis Overview	3
2	Related Work	3
2.1	EnCampus	5
2.2	Visualization Client Applet	5
2.3	Brooklyn College Registration System	5
2.4	Summary	8
3	Design Approach	9
4	Implementation	11
4.1	Calendar Query System	11
4.2	Timetable Query System	13
4.3	Room Availability Query System	15
4.4	Timeline Visualization System	17
5	User Evaluations	21
6	Conclusion	22
6.1	Summary	22
6.2	Future Work	23

List of Figures

1	Example course from the MyCampus website	2
2	Screen capture of EnCampus Timetabler	4
3	Example panel of the ViCA interface	6
4	Graph depicting scheduling patterns from Brooklyn College	7
5	Schedule information as displayed in MySQL	9
6	Data-flow diagram of the visualization system	11
7	Week view and course view for the calendar visualization	12
8	Drag and drop professors list	13
9	Timetable view displaying the schedule for a room	14
10	Room availability list example	16
11	Timeline room list visualization	18
12	Expanded room view with tooltip	18
13	Combined Visualization systems	19
14	List of values for timeline colours	20
15	Timeline building view	20

1 Introduction

The objective of this thesis is to design software to help users examine a schedule (of courses at UOIT) in such a way that they can determine which aspects of the schedule are acceptable or unacceptable. We hope the software designed can be used as a basis for a system capable of assisting the long-term needs of scheduling of the university. Visualizing a schedule can help point out underused resources (such as classrooms) in a large-scale schedule that may not have been identifiable in textual form. With a large amount of rooms available, there are numerous schedules that may be valid but visualization allows the user to assess factors like capacity and walking distance. A benefit to all staff and students is making sure a schedule is fair and balanced, so the primary focus can be on education and not attempting to overcome a difficult schedule.

1.1 Problem Statement and Motivation

The problem of managing resources such as rooms affects everyone in some form, this can especially be felt in learning environments where mismanagement of classrooms can adversely influence student learning. The large amount of simultaneous classes trying to be coordinated can result in situations where poor timeslots or inadequate rooms are used. These factors can play a role in not only the ability for students to learn but also the satisfaction of both students and faculty. Visualizing the schedule would allow those responsible for scheduling to make adjustments and improve resource allocation by giving them the necessary tools observe potential problem areas as well as potential solutions.

Intro. to Computer Science - 71260 - CSCI 1030U - 002							
Associated Term: UOIT Winter 2010							
Registration Dates: Jun 29, 2009 to Jan 22, 2010							
Levels: Undergraduate							
UOIT - Oshawa Campus Campus							
Laboratory Schedule Type							
0.000 Credits							
Registration Availability							
	Capacity	Actual	Remaining				
Seats	24	22	2				
Waitlist Seats	0	0	0				
Scheduled Meeting Times							
Week	Type	Time	Days	Where	Date Range	Schedule Type	Instructors
	Class	5:10 pm - 7:00 pm	R	University Building A1 UA3240	Jan 11, 2010 - Apr 16, 2010	Laboratory	TBA
** NOTE: This section is linked with CSCI 1030 - 001 (CRN: 71253).							

Figure 1: Example course from the MyCampus website

This is the only schedule information publically available for university courses at UOIT (image obtained from: http://www.uoit.ca/mycampus/avail_courses.html) [myc10].

1.2 Goals

The goal of this thesis is to design a prototype capable of presenting schedule information to all faculty and students at UOIT. To allow access to a large audience at UOIT the finished system should be web-based with a simple interface. The current system of displaying schedule information to students is not adequate to effectively relay information (see Figure 1). We believe an effective visualization of the schedule would directly benefit the UOIT community by helping students find study rooms, or help professors find more suitable room allocations for classes. The visualization software should openly present schedule information to everyone at UOIT in order to facilitate dialogue about improvements to current scheduling practices. This visualization software will be a stand-alone system and will not be integrated with existing university scheduling systems. The application will comprise two distinct, yet related components:

- Query-based systems that filter schedule information based on user input
- Visualization-based systems that assist the user in identifying problems in the schedule

1.3 Thesis Overview

In Section 2, we will investigate systems related to the projects goals in order to attempt to locate techniques that convey useful information for both those looking to improve the schedule, and those wishing to query information from it. Then in Section 3, the technique for implementing the system and justification for visualization choices will be discussed. Afterwards, in Section 4 the various intermediate and final tool versions will be discussed along with the reasons for different designs being required in order to achieve a framework closer to the goal of the thesis. User feedback given based on the final visualization-based system will be listed in Section 5. Finally, Section 6 will contain discussion about how well this system achieved the project goal, a summary of the thesis, and future work based on this design.

2 Related Work

Schedule visualization is a common technique that is frequently included with schedule software. For example, this is true for the software the university currently uses, Infosilem's EnCampus software (see Figure 2) [YMBW10] [Inf10b]. EnCampus features separate modules for: course scheduling, exam scheduling, student scheduling, and enterprise scheduling. Software packages like EnCampus help automate and simplify schedule creation as well as visualize the results. Non-commercial software shall also be examined that attempt to effectively display schedule data (the Visualization Client Applet) and then an examination of the registration system used at Brooklyn College.

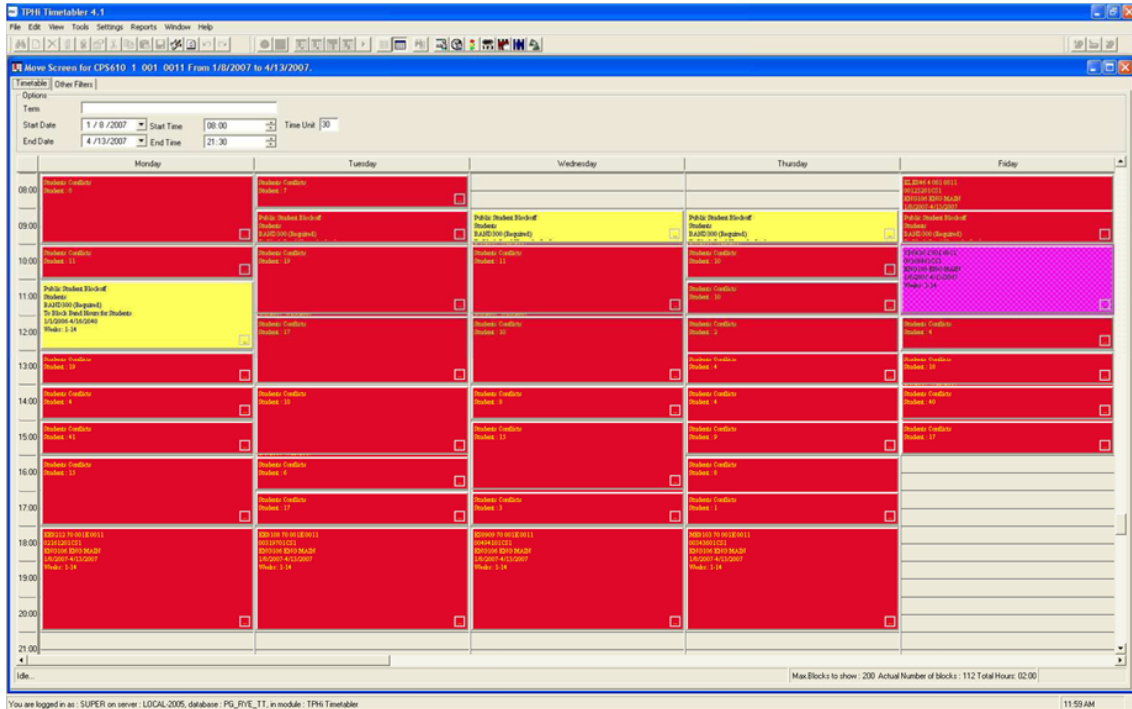


Figure 2: Screen capture of EnCampus Timetabler

Image obtained from:

[http://infosilemdownload.com/pdf_infosilem.com/Product\(space\)Leaflets/Timetabler.pdf](http://infosilemdownload.com/pdf_infosilem.com/Product(space)Leaflets/Timetabler.pdf) [Inf10a]

2.1 EnCampus

Other than high costs for some of these software packages (\$30,000 / year), the effectiveness of showing statistics like room utilization may vary. The visualizations generated by EnCampus are currently limited in mobility as the results of the program may not be viewable from remote locations. The strength of this kind of scheduling software is limited by lack of proper web-based accessibility, and is reliant on custom software to distribute information to a large audience.

2.2 Visualization Client Applet

Another example is ViCA (Visualization Client Applet) created by Kilponen and Vanhanen at the Helsinki University of Technology in Finland (see Figure 3) [KV00]. The visualization in ViCA is similar to the visualization in this thesis; however, it is based on using graphs and charts to display statistics and may not be suitable to handle information about scheduling. Additionally the paper on ViCA does not focus on what kind of data should be presented, but rather more theoretical problems such as how to gather data and how to display it [KV00].

2.3 Brooklyn College Registration System

A currently existing web portal that integrates schedule visualization involving tasks such as room availability is being used by Brooklyn College (see Figure 4) [GA04]. This system encompasses the goals of our project such as having an online system to allow easy access to the schedule data while also directly improving the scheduling. The Brooklyn College software operates within their registration software, which produces desirable results such as various reports and graphs

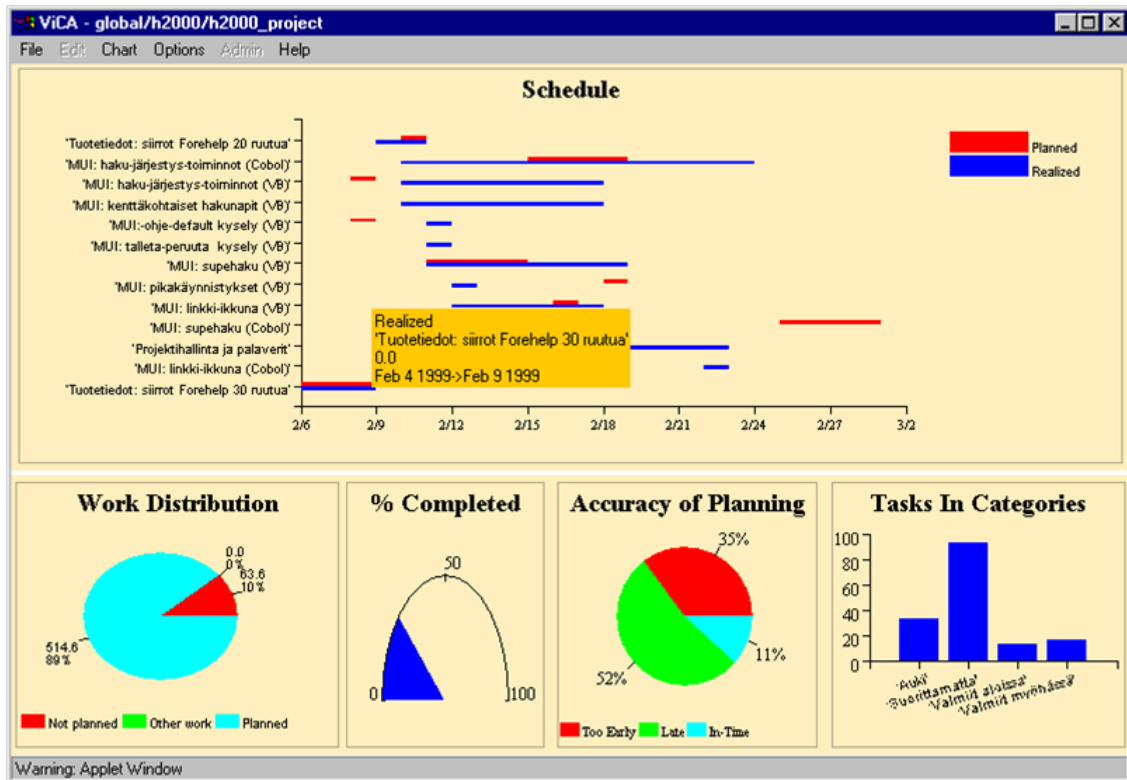


Figure 3: Example panel of the ViCA interface

Image obtained from:

<http://doi.ieeecomputersociety.org/10.1109/HICSS.2000.926944>[KV00]

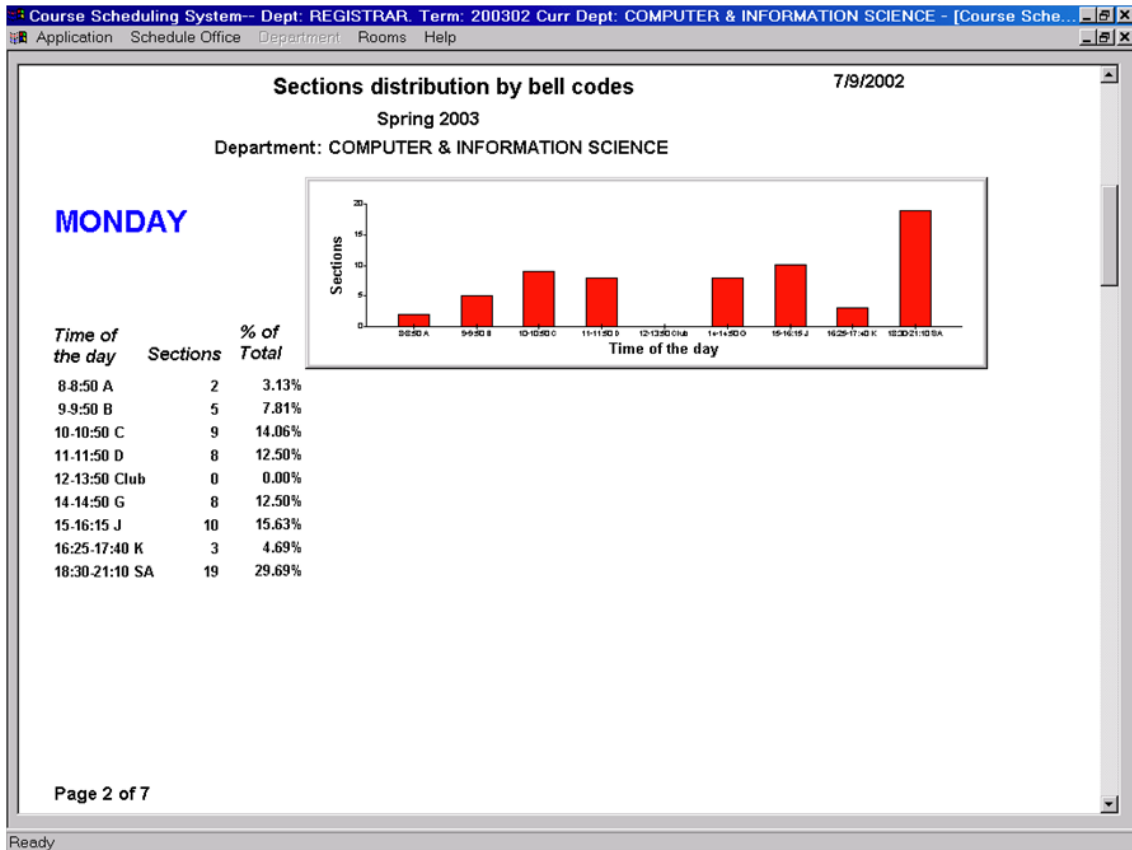


Figure 4: Graph depicting scheduling patterns from Brooklyn College

This figure is generated by the course scheduling system used by Brooklyn College (image obtained from: http://media.centerdigitaled.com/CUNY_04/The_Paperless_Registrars_Office.pdf)[GA04]

that "help ensure the most efficient and student-centered term schedule" [GA04]. The downside is that by integrating this program into the registration system, it limits the ability to make changes without impacting the visualization software. An example would be if the software used to register students was changed to a different commercial system, it would require the visualization code to be reintegrated each time the software was updated.

2.4 Summary

The ViCA screens show mostly graphs as a form of schedule information, missing more detailed designs like a timetable and instead presenting a more analytical interface, rather one that focuses on querying the schedule for specific information. This is likely due to its theoretical design, so it does not explicitly focus on the problem of location time-management. In order to allow a wide variety of users, the goal of this thesis is to have a system that is more focused on delivering information rather than statistics. Attempting to follow the ViCA approach would reduce the usefulness of the framework to an administrative or research tool as students and professors do not need information about percentage of student conflicts. The EnCampus software a larger variety of visualization methods than ViCA, such as the pictured timetable, while also feature general statistics such as graphs for course distribution that would be available in ViCA. This seems similar to the information that is also shown in the Brooklyn College web portal, which ideally the thesis framework would be capable of delivering. However, being able to successfully produce graphs of information and convey the schedule information are two different categories, and the focus of the thesis will be on visualization of the schedule information so that the system does not possess the problems of the ViCA system.

crn	day	startTime	endTime	building	room	code	crn	name	professor	type	capacity	maxCapacity
70001	4	12:40:00	14:00:00	University Building A3	UB2080	BIOL 1020U	70001	Biology II	DARIO T. BONETTA	Lecture	122	250
70001	6	15:40:00	17:00:00	University Building A3	UB2080	BIOL 1020U	70001	Biology II	DARIO T. BONETTA	Lecture	122	250
70002	2	17:10:00	18:30:00	University Pavilion	UP1500	BIOL 1020U	70002	Biology II	MARY OLAVESON	Lecture	198	250
70002	4	17:10:00	18:30:00	University Pavilion	UP1500	BIOL 1020U	70002	Biology II	MARY OLAVESON	Lecture	198	250
70003	3	09:10:00	12:00:00	University Building A3	UB4095	BIOL 1020U	70003	Biology II	TBA	Laboratory	22	24
70004	3	09:10:00	12:00:00	University Building A3	UB4095	BIOL 1020U	70004	Biology II	TBA	Laboratory	19	24
70005	4	09:10:00	12:00:00	University Building A3	UB4095	BIOL 1020U	70005	Biology II	TBA	Laboratory	24	24
70006	4	09:10:00	12:00:00	University Building A3	UB4095	BIOL 1020U	70006	Biology II	TBA	Laboratory	24	24
70007	5	09:10:00	12:00:00	University Building A3	UB4095	BIOL 1020U	70007	Biology II	TBA	Laboratory	20	24

Figure 5: Schedule information as displayed in MySQL

The format the MyCampus information is stored in the databases is shown here with the course and schedule information joined together. Days of the week are numerical starting at 2 for Monday, this is based on one of the two MySQL commands to return the day of the week as a number. Start times and end times for classes are converted in the MySQL time format so information such as the hour can be retrieved by using MySQL functions.

3 Design Approach

The approach to designing the system began with deciding on the data necessary for simple schedule visualizations. The required information was accessible through the UOIT course preview page ([myc10]), which presents the data as a HTML file containing all available courses for the Fall 2009 semester. This page was manually downloaded. In order to make the visualizations possible, all schedule information had to be removed from the file and placed into a database. This was done by creating a program that used regular expressions to extract all relevant information from the HTML file and produced two comma-separated values (CSV) files:

- One file containing the information about each course (unique CRN)
- One file containing all scheduled class times occurring that semester (non-unique CRN)

These two files were then used to create two SQL tables, which could be queried in order to generate visualizations of the schedule, additional information (e.g., room capacity) that was required but not in the course preview page was added to the tables manually. The format of the database is shown in Figure 5.

The web-based software was created in PHP so that it could connect to the SQL tables and automatically generate visualizations. The load time of visualizations was optimized through caching, although most of the visualizations did not have long generation times with the exception of the final room usage visualization-based system, which can take a few seconds to load. Since each page is dynamic, all visualizations could be updated instantly after the data tables are updated, this would be useful if the system had direct access to the information being updated. The integration with the existing university databases seems straightforward with minor modifications to the existing code.

The platform to deliver the visualization is web-based to allow staff and students the opportunity to access the final product and possibly elicit feedback during the prototyping phase. The designing phase consisted of attempting to fill in the missing gaps of information, and answer questions based on the data that was available from the schedule. Desired information consisted of topics such as:

- Generating the current schedule of a student/professor
- Listing all the classes occurring in a classroom
- Identifying courses of different faculty
- Comparing schedules for two different students/professors

The dataflow of the final system is represented in Figure 6.

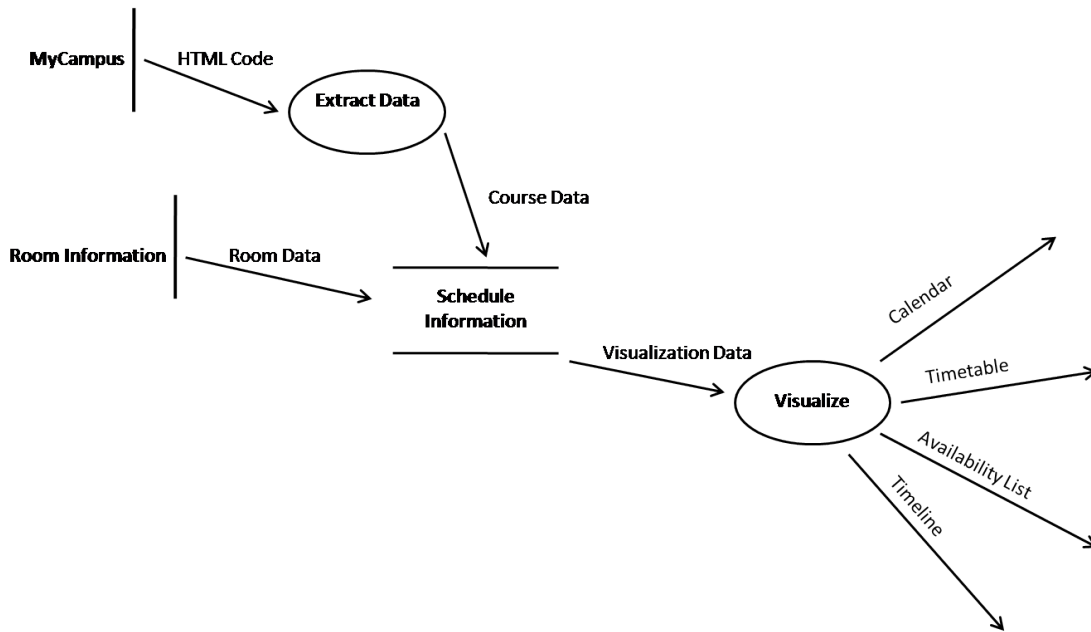


Figure 6: Data-flow diagram of the visualization system

4 Implementation

4.1 Calendar Query System

The first stage of attempting to display the data resulting from the different filters started as returning rows of data from a MySQL client. This was not a suitable delivery method for the schedule information due to the difficult readability (see Figure 5), so a design that would be more appropriate to display time relative data was required. The first system was a calendar that could sort query results into the different days of the week. The project used modified open-source software for this phase in order to decrease implementation time while featuring suitable functionality to handle delivering all the necessary information to implement the currently desired filter types. Existing web-based calendars such as Google Calendar [Goo09] and the TikiWiki calendar [tik09] were examined in order to determine if they could be adapted for schedule visualization. However, they were

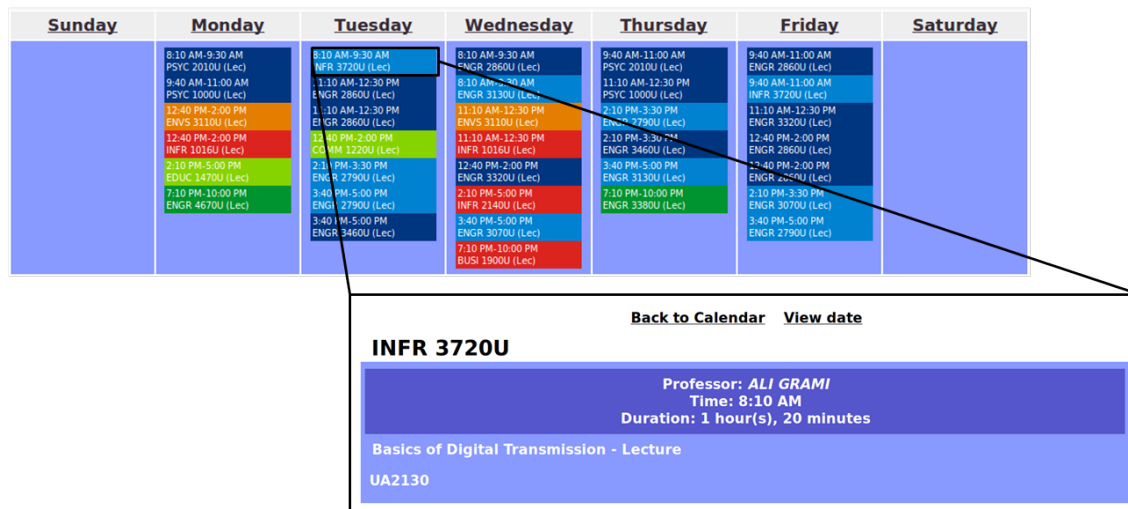


Figure 7: Week view and course view for the calendar visualization

This figure was generated using the web form in Figure 8 to select multiple professors. The week view (top) is the result of the query, and the course view (bottom) is the result of click on one of the classes. The colours of the boxes each correspond to a different professor teaching the class.

limited in customization and too complex to modify for the purpose of this thesis. The starting software to adapt to the visualization design was a web calendar called PHP-Calendar [php09]. It had a good set of features such as event boxes that linked to more detailed information. Using this open-source software as a starting point, the calendar was modified to turn the calendar from an event scheduler into more of a table of results for filtering schedule information. The calendar went through a large number of iterations to allow most of the initial list of query searches to function; features such as clickable classes allowed information like course information to be displayed without cluttering the calendar. In Figure 7 is a picture of the week view and course view that was generated using the calendar visualization. All of query-based systems use a type of HTML form ranging from text boxes to more complex interactive forms created in jQuery. The form used for selecting professors is shown in Figure 8.

Eventually the calendar reached a stage of improving user interface features such as using

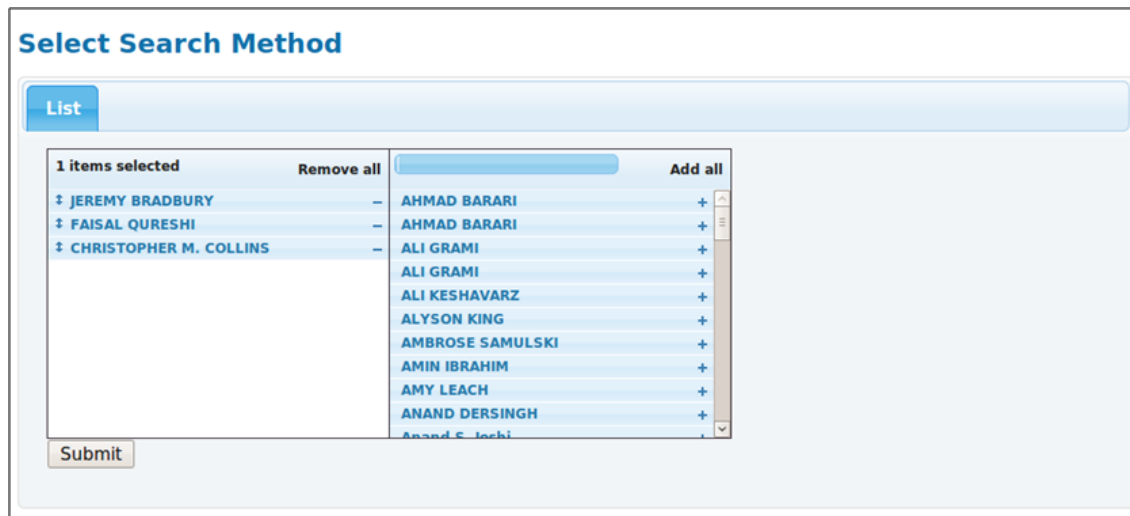


Figure 8: Drag and drop professors list

This form contains a list of professors (right) that can be added to the left side in order to query for several professors at once. It is used with the calendar and timetable visualizations.

colours to differentiate different professor classes. Around this time, the inability of the calendar to represent times in graphic form was limiting the ability to make it an effective manner of visualizing aspects like daily events. The calendar displayed a day as a single column list of classes without any spatiality, which made comparing hourly usage of rooms, or overlap in schedules difficult.

4.2 Timetable Query System

The next iteration of the visualization designs was a timetable; this would allow classes to be displayed in a table where rows represented time in the day and columns represented the days. The timetable was designed using an HTML table in order to mimic the currently existing timetable available to students on MyCampus. By having the rows spaced and sized relative to their starting time and duration, it allows easy visualization of query results such as availability of a room or volume of classes in a student schedule. The large boxes used to represent classes allowed for

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00 am					
		COMM 1320U CRN: 71279 Lecture 8:10 am - 9:30 am UL12			
	ENGR 4500U CRN: 71599 Tutorial 8:40 am - 10:30 am UL12				
9:00 am					
				COMM 1320U CRN: 71278 Lecture 9:40 am - 11:00 am UL12	COMM 1320U CRN: 71279 Lecture 9:40 am - 11:00 am UL12
10:00 am			CHEM 4420U CRN: 70114		

Figure 9: Timetable view displaying the schedule for a room

information such as start time, end time, CRN, class type, and professor to fit easily inside the box without the need of additional windows as seen in Figure 9.

The problem with using a HTML table to generate the timetable was that it could not handle overlapping classes. This made the timetable view more limited than the calendar view in terms of the number of query results that could be displayed. For example, the timetable could not display multiple professors at the same time, as there was no way to handle overlapping classes. So instead it was used in combination with the calendar, and displayed whenever a query could not result in overlap (such as room schedules). Given enough time, it would have been ideal to rewrite the timetable to use stylized boxes instead of table cells in order to allow all filters capable of being displayed on the calendar to be transferred to the timetable view. With the combination of the calendar and timetable to display schedule information, the thesis changed focus to other methods of visualizing the data and queries that could not be filtered through the current design. A relevant query that could not yet be answered was how to located empty rooms in the UOIT schedule.

4.3 Room Availability Query System

The calendar and timetable view cannot locate empty rooms because they do not have the methods to display them properly, and the query is too complex for their design. Both the calendar and timetable depend upon taking a row from the SQL tables and converting it into a box displaying information about a course. There is no way to make a box representing data such as an empty room because it would require information such as course code, professor, start time and end time which does not corresponding to an existing table entry. In order to make the old visualization methods work, it would require the tables to be populated with data for empty rooms. This process would increase the size of the schedule tables significantly, and would be unnecessary because this data is already derivable based on the existing data.

In order to locate empty classrooms and other relevant information like the time of the next class, a new webpage was designed to print rows of data about empty classrooms for a given duration (see Figure 10). In order to avoid the problem of making "available classroom" entries in existing tables, the information for availability period for each classroom is obtained by returning gaps between classes in the Schedule table. This is then supplemented by any other relevant data such as the building the room takes place in, or the capacity.

The room availability webpage gives the user a sufficient amount of information, but a time is required input. A user could attempt to adjust time and see the resulting changes, however this is time consuming and there is no real way to view the information without a predefined starting point. This list is difficult to tie to the original goal of visualizing an entire schedule when it only presents a very limited view. This view would make a great tool but would have to be used in combination with other visualizations. Up until this time, the calendar, timetable, and room availability

Rooms available on Monday from 8:10 am to 11:00 am			
Building	Room	Availability Period	Capacity
B-Wing	B305	8:10 AM - 10:00 PM	
C-Wing	C212	8:10 AM - 10:00 PM	
D-Wing	D214	8:10 AM - 10:00 PM	
EDUCATION CENTRE	EDU212	8:10 AM - 4:10 PM	30
EDUCATION CENTRE	EDU311	8:10 AM - 4:10 PM	30
EDUCATION CENTRE	EDU313	8:10 AM - 4:10 PM	30
EDUCATION CENTRE	EDU410	8:10 AM - 10:00 PM	45
F-Wing	F128	8:10 AM - 11:10 AM	
Georgian	College	8:10 AM - 10:00 PM	
Georgian College	H209	8:10 AM - 10:00 PM	
H-Wing	H166C	8:10 AM - 10:00 PM	
H-Wing	H221	8:10 AM - 12:40 PM	
I-WING	I210	8:10 AM - 10:00 PM	
L-Wing	L114	8:10 AM - 10:00 PM	
L-Wing	L118	8:10 AM - 10:00 PM	
L-Wing	L212	8:10 AM - 10:00 PM	

Figure 10: Room availability list example

This figure is displaying rooms that are empty on Monday from at least 8:10 am to 11:00 am.

list focused on delivering results to specific queries. None of these previous visualizations gave the user any way to browse the available schedule information freely. There were no general statistics or simple dashboards to see what is contained in the schedule. This restricted the usefulness of the system as it puts the onus on the user to find relevant information themselves rather than the webpage freely displaying this information, as it should. After the empty room list was created, there was brainstorming of other visualization methods such as geographical maps displaying capacity information for rooms, but the amount of work required made commitment to this design difficult. Instead, a new visualization method was created, allowing information about capacity to be displayed while also having the simplistic overview that allowed information like empty rooms to be examined implicitly.

4.4 Timeline Visualization System

Up until this point, the calendar was still the system most capable of displaying all the schedule data because the timetable could not handle that much data and the non-graphic form of the room availability list was difficult to examine quickly. However, the calendar system had flaws as discussed earlier and required improvement, so a new visualization-focused system was designed. The next step in visualization of the UOIT schedule was a system that would display a large amount of information for each building, without the need to narrow down any excess information through filters. This system was mainly oriented to the visualization side of sharing schedule information, unlike the previous system that focused on forming queries to filter the data (and therefore this system is referred to as visualization-based). The main focus was on room occupancy, which would be displayed through a bar with various shades of blue boxes depicting the occupancy percentage.

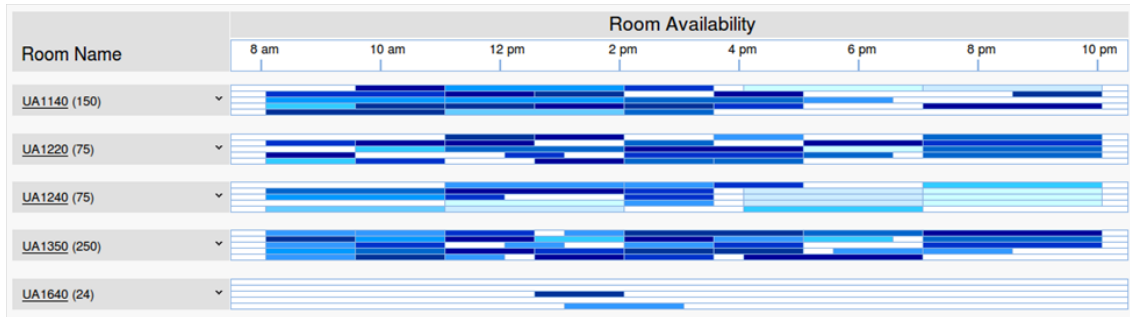


Figure 11: Timeline room list visualization

This figure displays the default room list view for selecting University Building A1 from Figure 15. For a legend of the colours, see Figure 14.

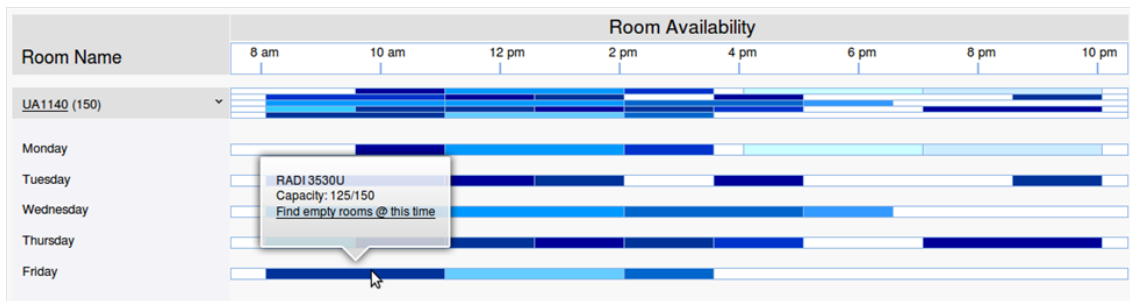


Figure 12: Expanded room view with tooltip

The position and length of the bars were equivalent to their start time and duration respectively. Each room has up to five bars, with each bar representing a day of the week. The rooms are listed down the left side of the page with the room capacity listed in brackets beside each room name and a small version of their room occupancy bar is beside each room (see Figure 11). The arrow to the left of the visualization bars expand a menu that shows larger bars with listings of the days they represent. These larger bars also have tooltips that appear when hovered over, they list information such as course code, course capacity, room capacity, and a link to the room availability list (see Figure 12). Each link to the room availability list displays the rooms available for the length of the selected class; this is to help for tasks like finding alternate rooms for a class. Additionally the

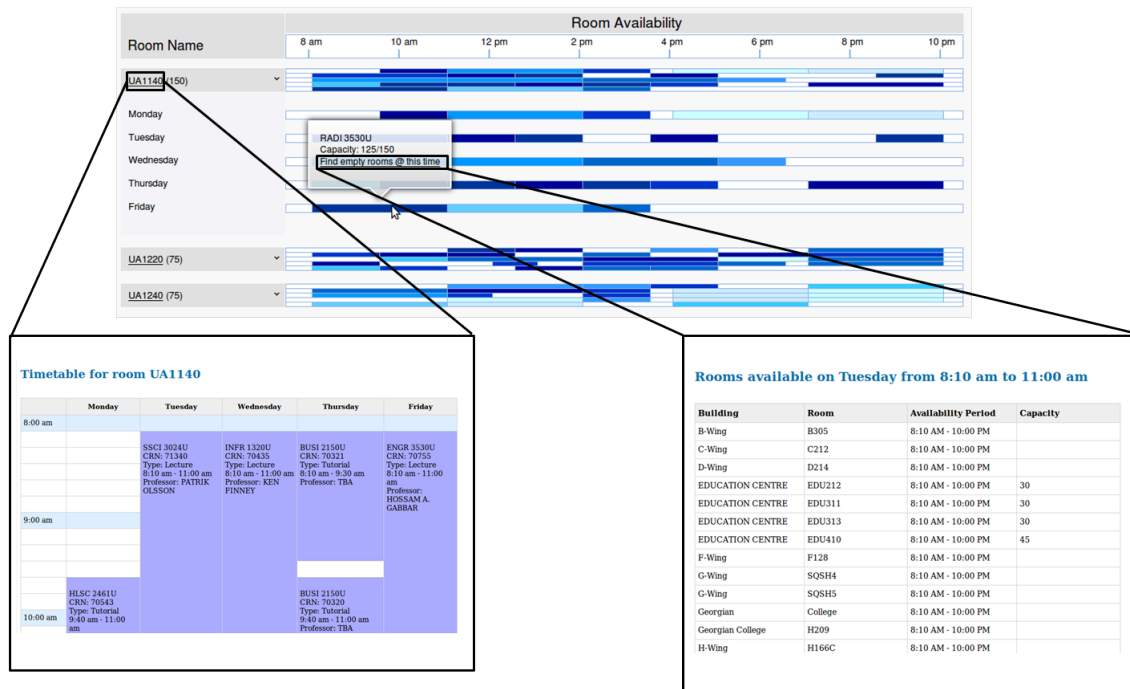


Figure 13: Combined Visualization systems

This figure shows the Visualization-based system (top) with the links to the older query-based systems, the timetable (bottom left) and room availability list (bottom right).

names of each room on the left side of the screen also act as links to the timetable view (see Figure 13). This helps show class occupancy in a more traditional method and gives the user access to more data such as the professor of the class.

The bar colours are broken up into 10 different shades of blue. This is because the human eye has trouble distinguishing between finer variations of colour. Each shade of blue represents increments of ten percent of the room capacity, any value over room capacity is listed in red, and rooms without any current capacity information are represented as grey (see Figure 14). In order to access the list of rooms, a page displaying a list of the buildings was created in a similar method. Each building was represented by a group of bars, where each bar represented the average capacity for a single room as shown in Figure 15.

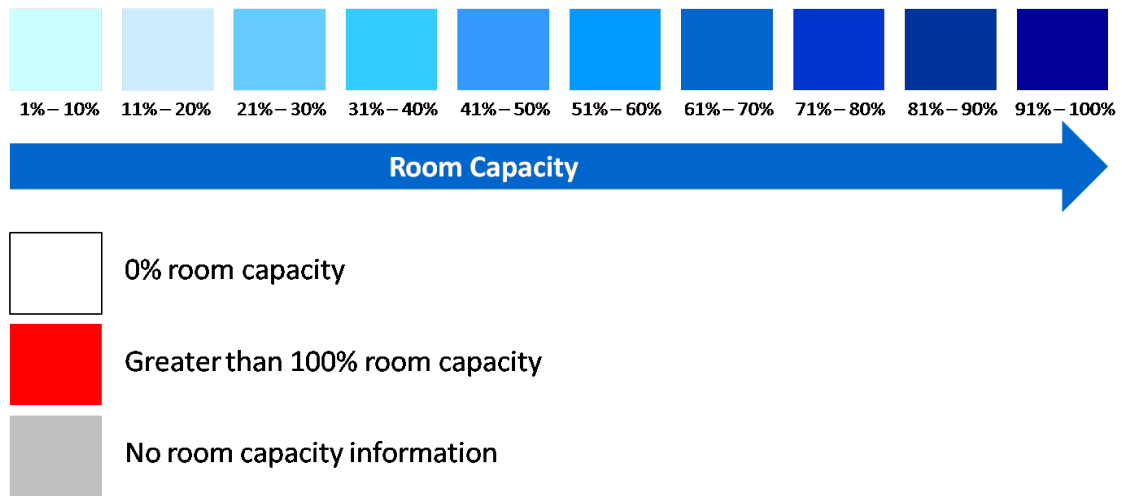


Figure 14: List of values for timeline colours

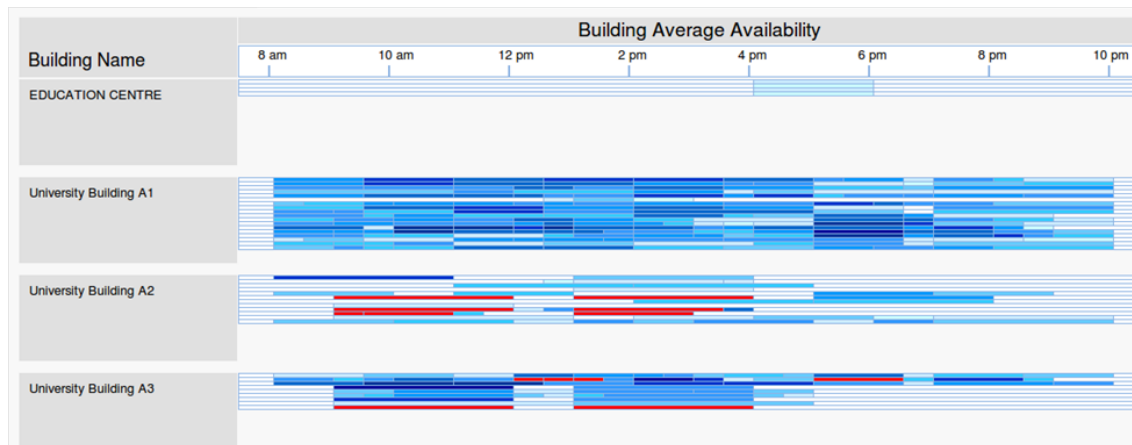


Figure 15: Timeline building view

A portion of the building view for the visualization system; acts as a start page for accessing lists of rooms by building.

5 User Evaluations

After the visualization-based system was finished, other tasks such as presenting the current system to scheduling faculty at UOIT were performed to elicit feedback [YMBW10]. The user evaluations with the scheduling faculty consisted of a presentation about the visualization software and a brief video demo. The meeting gave a chance to hear suggestions about how the system compares to the UOIT software and how the thesis framework could be improved and potentially utilized by the scheduling department. While the current Infosilem software being used is a complete package for schedule visualization, there was lamentation about the ability to share the information, particularly a web-based solution was missing. It was discussed that the current solution is a simple system created by the school, which the thesis webpage could potentially relieve some of the system's shortcomings. This included problems with readability, an example being that classes are separated into half-hour chunks. Comments about the thesis webpage also mentioned being used as a "dashboard" [YMBW10] to give a quick glimpse of the current schedule for presentation purposes. The presentation of the thesis software brought up lots of discussion about attempting to create a collaborative system that combined projects like this and other interested parties with the existing data. From this lots of discussion of the potential possibilities that would surpass some existing difficulties in the schedule visualization software; these will be discussed in the future work section. Overall, all discussion was either positive or related to potential improvements that went beyond the scope of this thesis, such as integrating student classes in the visualization-based system. Even if the impact of this thesis is minor, its framework could help the scheduling department with analyzing already created schedules, and potentially help make the initial steps of schedule creation shorter and simpler.

6 Conclusion

The final designed visualization based system helped convey information such as capacity and general room availability easily at a glance. When paired with the timetable and room availability list the system became a powerful tool that helped a user browse for interesting scheduling occurrences. The designed system did not have any major flaws and since the remaining time for designing the visualization software was brief, the webpage underwent a feature freeze at this point.

6.1 Summary

The designed visualization system went through several iterations, each with its own shortcomings. After lots of discussion, a visualization-based system was designed that helped tie together previous query-based systems into a cohesive framework. The existing system achieves the goal of effectively visualizing the room schedule of UOIT, and acts as a good starting point for future refinement and expansion on the topic of schedule visualization. While it is unlikely that any designed system based on the current visualization webpage could replace commercial products, it could produce an effective and cheap solution portal for faculty and staff to better understand the existing schedule. Additionally it could be the starting point for creating a proper collaboration between the scheduling staff and student projects for creating an ideal system that could be integrated more deeply into the existing school online portals.

6.2 Future Work

The next step in the thesis will be to host the existing web pages on a server that can be publically accessed. This will allow the computing science faculty of UOIT to test the software, and determine if it should be publically used. The current system lacks data that is not related to the university, which unfortunately does not encompass the entire schedule visualization problem. Schedule information for Durham College, Trent University, night classes, and onetime events would help visualize the entire campus and help fill in missing gaps of information. Additionally the current system is based on creating data for a single week, this causes problems for biweekly classes and special events, so integrating data that is less static would be a necessary step to improve accuracy.

Other possible improvements cover things outside the scope of this thesis such as student schedules. This addition would make the system capable of more advanced visualizations such as quickly creating a student's timetable, or showing general graphs about average class times for students. These changes would also make the system competitive with the existing scheduling software, and allow those in the scheduling department another tool to help increase the speed of creating the school schedule. Ultimately, this system could be used as the framework for a very advanced system capable of displaying a large variety of information, it is important to work towards a system that fulfills the needs of both the students and the faculty at UOIT.

References

- [GA04] Mark Gold and Joan Antonicelli. The paperless registrar's office. Technical report, Brooklyn College, November 2004.
- [Goo09] Google. Google calendar apis and tools. <http://code.google.com/apis/calendar/>, 2009.
- [Inf10a] Infosilem. Infosilem encampus timetabler. http://infosilemdownload.com/pdf_infosilem.com/Product%20Leaflets/Timetabler.pdf, 2010.
- [Inf10b] Infosilem. Scheduling for higher education. <http://www.infosilem.com/infosilem.com/pages/Homepage/homepage.php>, 2010.
- [KV00] Pekka Kilponen and Jari Vanhanen. Vica - a www based tool for visualizing project status. <http://doi.ieeecomputersociety.org/10.1109/HICSS.2000.926944>, 2000. Hawaii International Conference on System Sciences.
- [myc10] Class schedule search. http://www.uoit.ca/mycampus/avail_courses.html, 2010.
- [php09] Php-calendar. <http://www.php-calendar.com/>, 2009.
- [tik09] Tikiwiki cms/groupware. <http://info.tikiwiki.org/tiki-index.php>, 2009.

[YMBW10] Kimberley Young, Brad MacIsaac, Bruce Bunker, and Michelle Warrington. Interview. Electronic Recording, April 2010.