10-1996

Transit Time Internet Access: Prototype and Progress

Janet Vorvick
Portland State University

Transit Time Internet Access:
Prototype and Progress

by Janet Vorvick

October 1996
PR099

This project was funded by Tri-Met and Transportation Northwest (TransNow) Regional Center under the sponsorhip of the Department of Transportation UTC Grant Program. The U.S. Government and Tri-Met assume no liability for the contents or use thereof. The contents do not necessarily reflect the views or policies of TransNow, the U.S. Department of Transportation, or Tri-Met.

Center for Urban Studies
School of Urban and Public Affairs
Portland State University
Portland, OR 97207-0751
(503) 725-4020
(503) 725-5199 FAX
http://www.upa.pdx.edu/CUS/

Table of Contents

Abstract

Transit Time Internet Access Version 1 (TTIA1) is a prototype messaging program which delivers real-time bus schedule information to users of the Internet. Using the World Wide Web and the well-developed hypertext markup language (html), TTIA1 allows a bus user to request and receive schedule deviation information about a specific bus at a specific timepoint. TTIA1 is part of a project whose goal is to evaluate the effect on riders behavior and riders level of satisfaction of actual arrival time information.

For a demonstration see http://www.upa.pdx.edu/~janetv/TTIA/OUTPUT/top.htm

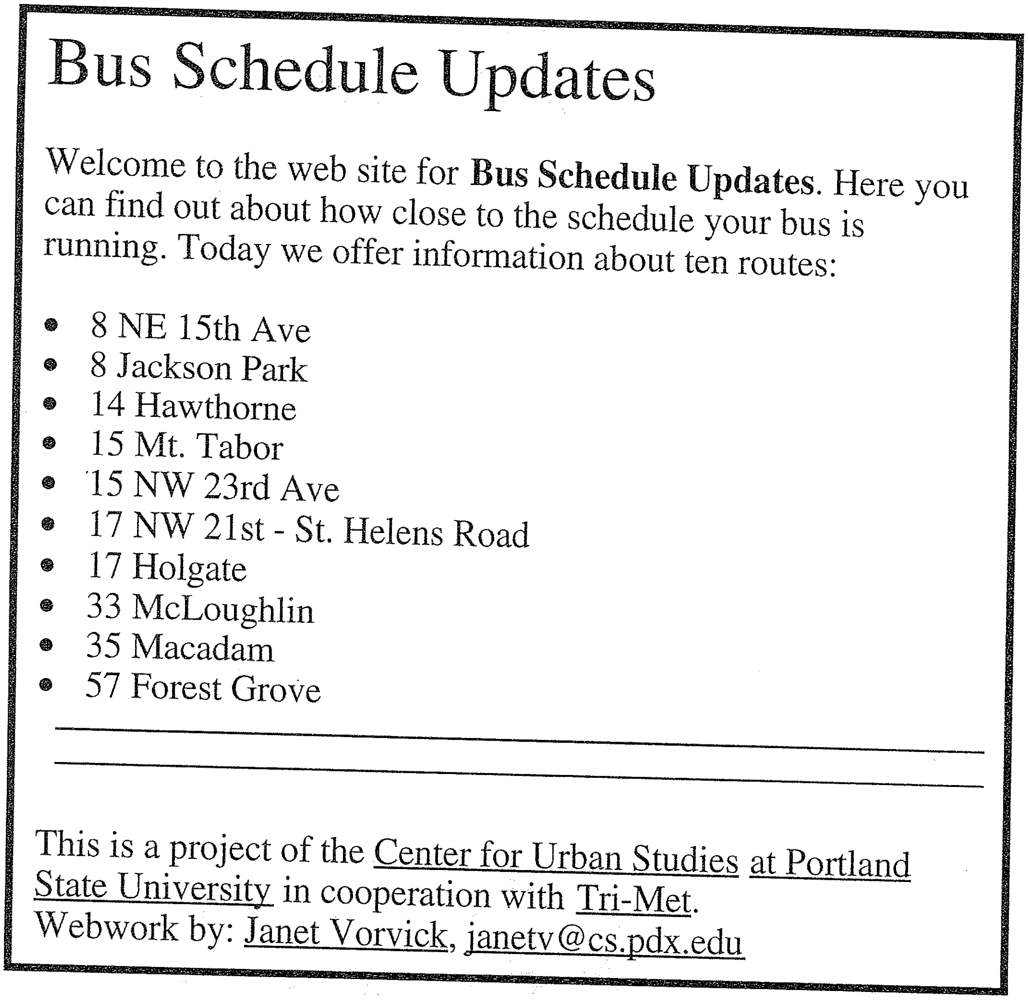
Introduction

Transit Time Internet Access Version 1 (TTIA1) consists of a group of programs which allow a user of the bus system of Tri-Met, the Tri-County Metropolitan Transportation District of Oregon, to access real-time information about the status of their bus with regard to its schedule. The user interface for TTIA1 is provided by the user's Web browser. The data for the TTIA1 program comes from Automated Vehicle Location (AVL) devices on board Tri-Met buses.

A user of TTIA1 receives an actual arrival-time estimate by opening an html document. This corresponds to clicking on a link on a Web page. Assisting the user, TTIA1 offers a series of four choices that determine which bus is the user's bus-of-interest and at what location they plan to board. The schedule update page for the user's bus-of-interest reports either that the bus is early, is late, is running on time or that there is no information available for the bus-of-interest. Proposed extensions of TTIA1 would report reroute information and also alert the user if the bus-of-interest is out of operation.

The Look and Feel of TTIA1

For an experienced user of the World Wide Web, TTIA1 is very user-friendly. Users who haven't experienced the Web have to learn only the actions that any Web user learns: point and click to open a page and bookmark those pages used often. The top page of TTIA offers a choice of ten routes:



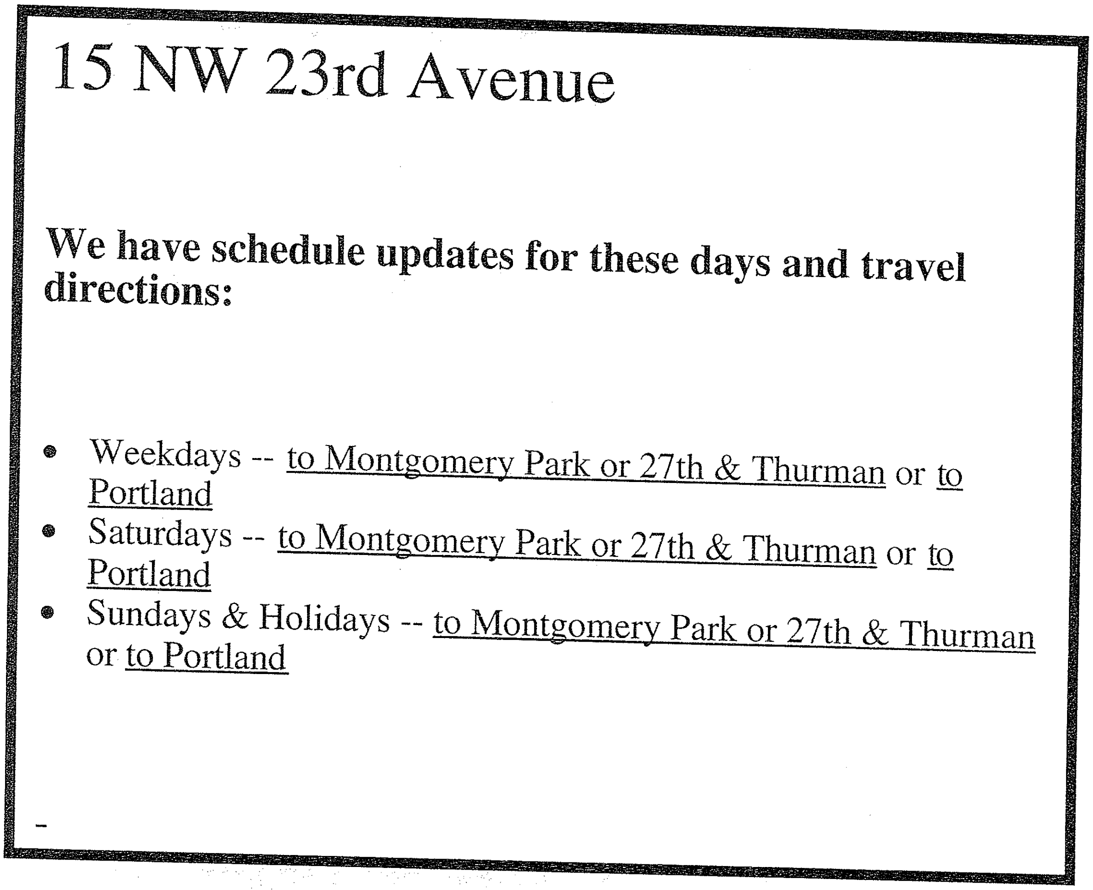
Bus Schedule Updates

Welcome to the web site for **Bus Schedule Updates**. Here you can find out about how close to the schedule your bus is running. Today we offer information about ten routes:

- 8 NE 15th Ave
- 8 Jackson Park
- 14 Hawthorne
- 15 Mt. Tabor
- 15 NW 23rd Ave
- 17 NW 21st - St. Helens Road
- 17 Holgate
- 33 McLoughlin
- 35 Macadam
- 57 Forest Grove

This is a project of the Center for Urban Studies at Portland State University in cooperation with Tri-Met.
Webwork by: Janet Vorvick, janetv@cs.pdx.edu

After making a choice of a route about which they want real-time information, the user is offered a choice of days and directions for the route:

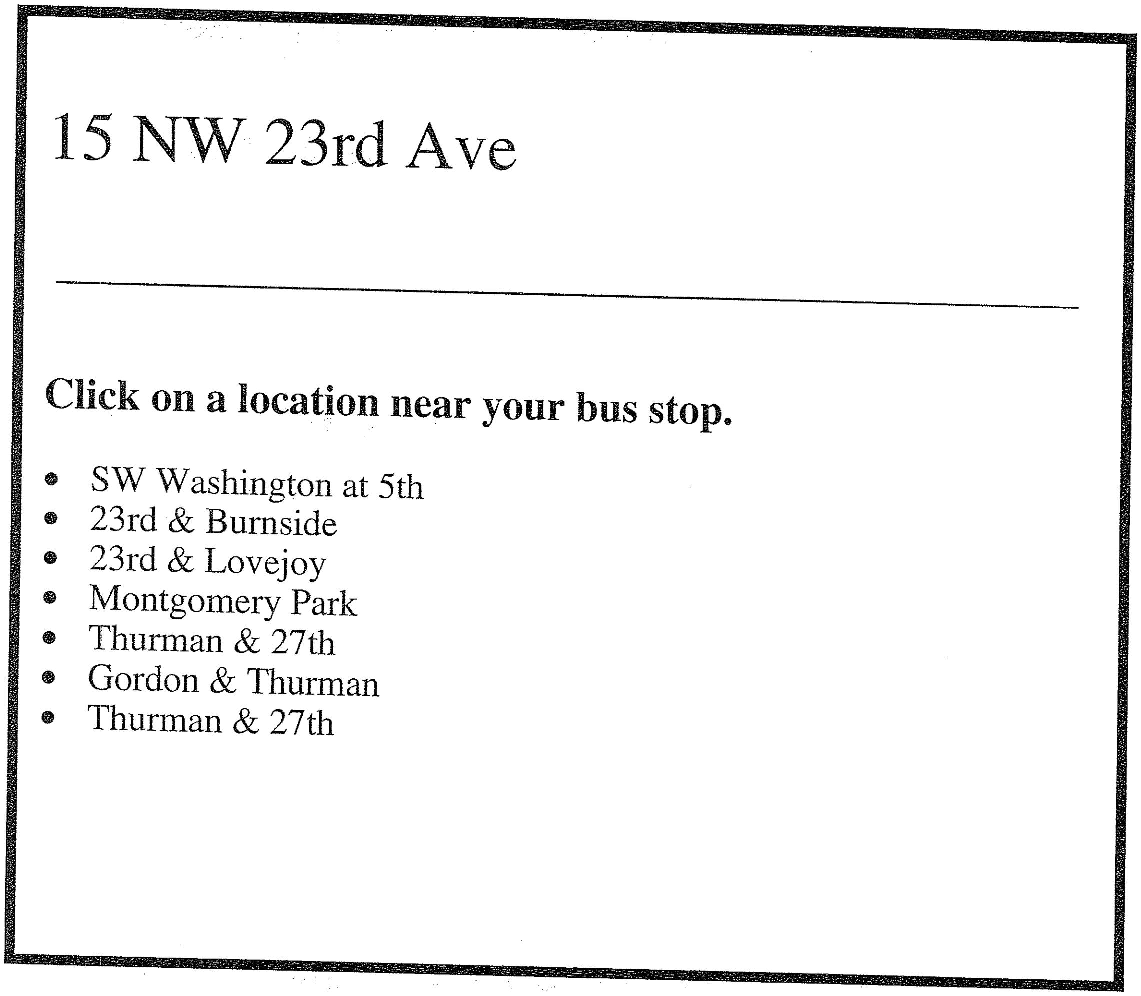


15 NW 23rd Avenue

We have schedule updates for these days and travel directions:

- Weekdays -- to Montgomery Park or 27th & Thurman or to Portland
- Saturdays -- to Montgomery Park or 27th & Thurman or to Portland
- Sundays & Holidays -- to Montgomery Park or 27th & Thurman or to Portland

Subsequent to their choice of day and direction, the user is asked to select a bus stop:

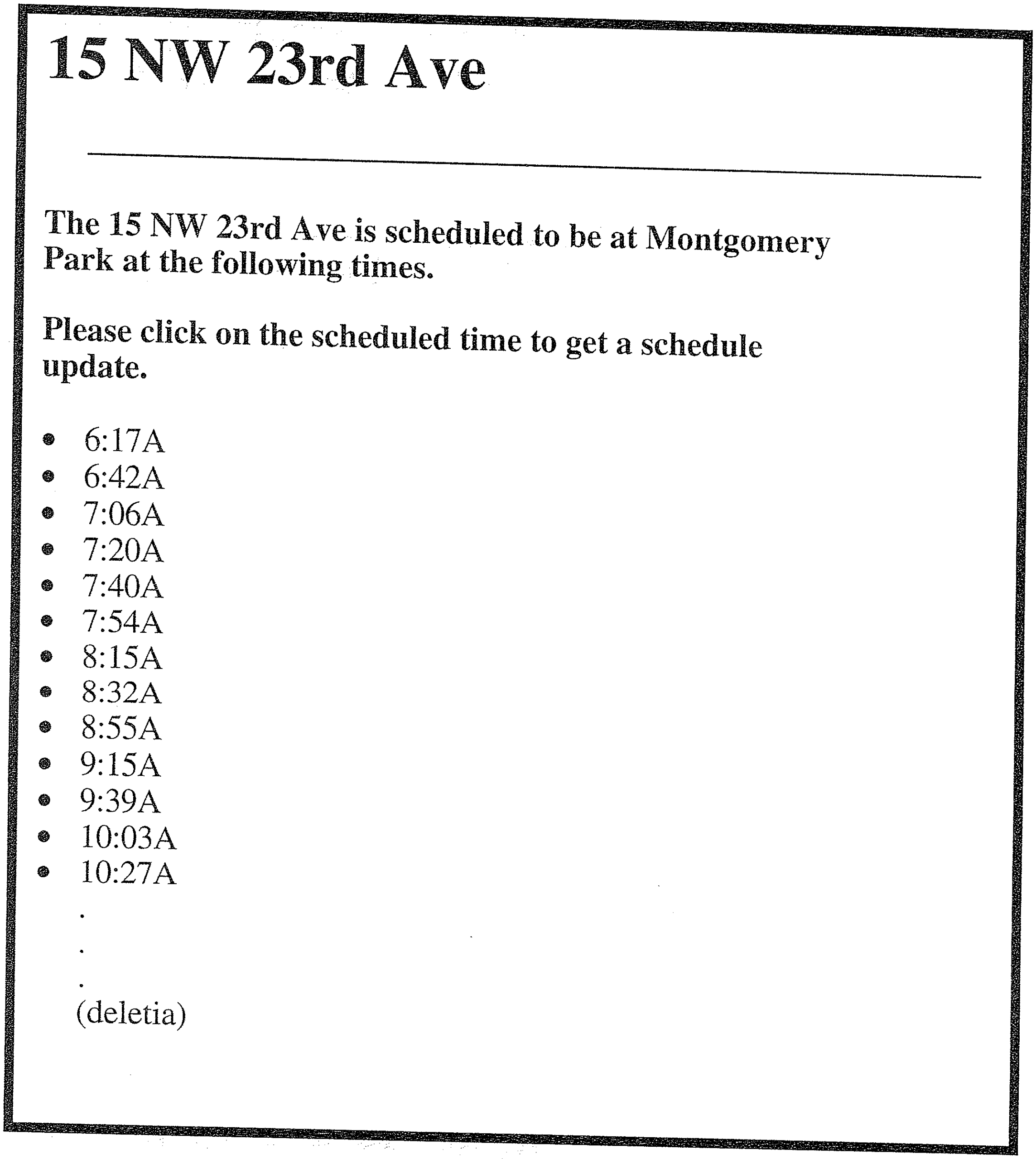


15 NW 23rd Ave

Click on a location near your bus stop.

- SW Washington at 5th
- 23rd & Burnside
- 23rd & Lovejoy
- Montgomery Park
- Thurman & 27th
- Gordon & Thurman
- Thurman & 27th

Finally, the user needs to specify the scheduled time they are interested in. The result of clicking on that time will be the presentation of the update for that bus.



15 NW 23rd Ave

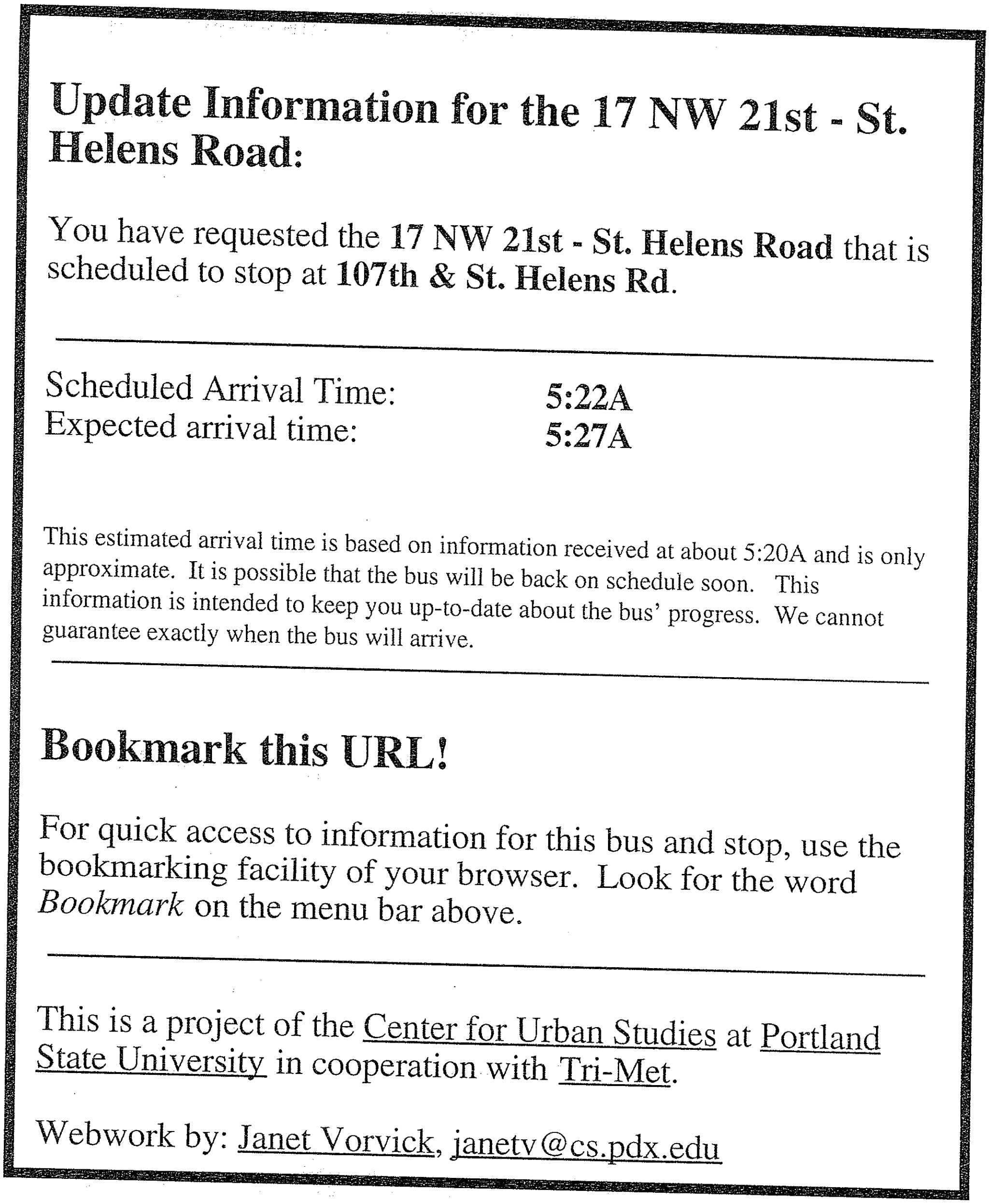
The 15 NW 23rd Ave is scheduled to be at Montgomery Park at the following times.

Please click on the scheduled time to get a schedule update.

- 6:17A
- 6:42A
- 7:06A
- 7:20A
- 7:40A
- 7:54A
- 8:15A
- 8:32A
- 8:55A
- 9:15A
- 9:39A
- 10:03A
- 10:27A

.
.
.
(deletia)

The update page offers advice and disclaimers.



Update Information for the 17 NW 21st - St. Helens Road:

You have requested the **17 NW 21st - St. Helens Road** that is scheduled to stop at **107th & St. Helens Rd**.

Scheduled Arrival Time: **5:22A**
Expected arrival time: **5:27A**

This estimated arrival time is based on information received at about 5:20A and is only approximate. It is possible that the bus will be back on schedule soon. This information is intended to keep you up-to-date about the bus' progress. We cannot guarantee exactly when the bus will arrive.

Bookmark this URL!

For quick access to information for this bus and stop, use the bookmarking facility of your browser. Look for the word *Bookmark* on the menu bar above.

This is a project of the Center for Urban Studies at Portland State University in cooperation with Tri-Met.

Webwork by: Janet Vorvick, janetv@cs.pdx.edu

If the bus is running on time, a different message appears:

> Scheduled arrival time: 5:22
> Expected arrival time: 5:22
> The bus has not reported any deviation from its schedule.

A third message is sometimes delivered:

> There is no information available about the bus at this time.

This third message differs from the 'on time' message because any request for actual arrival time information which is made too early will be meaningless.

The advice given on the update page about bookmarking the URL (Universal Resource Location, usually a web page) helps users avoid repeating the four page selection process each time they wish to check on their bus' progress. In addition, a user can expedite the selection of other routes and times by bookmarking the TTIA1 Welcome page.

The flow of the TTIA1 Web pages reflects some assumptions about the users. TTIA1 is intended for use by bus riders who have direct and continuous access to the Internet. Consequently, users are more likely to check on the status of a bus they ride from their downtown office toward home than they are likely to check on the status of a bus they ride from their home toward downtown. A downtown office is more likely to have an on-going link to the Internet. Though the information delivered by TTIA1 would be useful at the bus stop (perhaps at a kiosk), TTIA1 is intended for use by a bus rider in his or her own home or office.

Many kind and cooperative volunteers have made suggestions about TTIA1. That the information would be useful at the bus stop was often mentioned. Expansion of the choice of locations to include all the bus stops on the route - currently only the timepoints listed on the printed schedules are displayed - was a common request. Users are also interested in help deciding when to leave to catch the bus. This would require TTIA1 to ask the user to enter the amount of time they need to walk to the bus stop.

These and other aspects of the look and feel of TTIA1 were influenced by the usual constraints of research projects: time constraints, data availability, the parameters of the research goals, talents of the researchers, etc.

Description of the System - General

The TTIA1 system consists of two parts: the user interface and the update generating program. The user interface can be provided by any Web browser. The user of TTIA1 does not obtain any special software for running TTIA1. He or she simply uses the Web browser they (probably) already have on their computer. Netscape and Microsoft Internet Explorer are popular Web browsers. Several free Web browsers are available.

Web browsers access documents that are specially formatted. All the TTIA1 documents use the hypertext markup language (html) as their format. Browsers can find specific html Web pages by decoding their names. Also, they can display those pages in an attractive form regardless of the size of the window they are operating in. Thus html was the choice for the TTIA1 pages.

The TTIA1 html documents are stored on a computer at Portland State University. The users of TTIA1 access those documents temporarily. Some of the TTIA1 pages are static - they have no text that needs to change. Others, specifically the update report page, have to be created with real-time data. About 600 pages reside on the computer which runs the TTIA1 program. More than 30,000 links are needed for the routes and times.

The computer resources needed to *use* the TTIA1 system are these:

- a Web browser
- access to the Internet

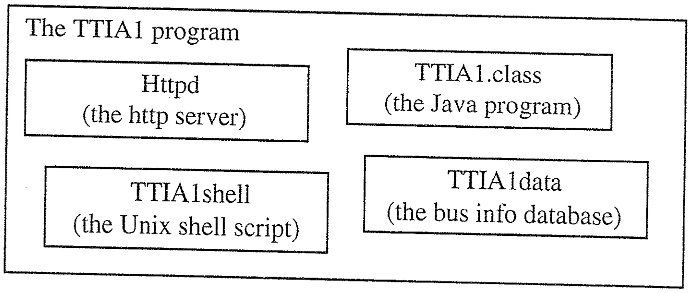
The computer resources needed to operate TTIA1 (or *any* TTIA1-like program) are these:

- direct and continuous access to the Internet (for delivering Web pages)
- an http server (delivers html documents to browsers)
- direct and continuous access to real-time bus schedule information
- standard mechanisms to compile and execute programs

The computer resources needed to run the *actual* TTIA1 program are these:

- the Unix operating system
- the Java programming language

TTIA1 is composed of four parts, as pictured below.



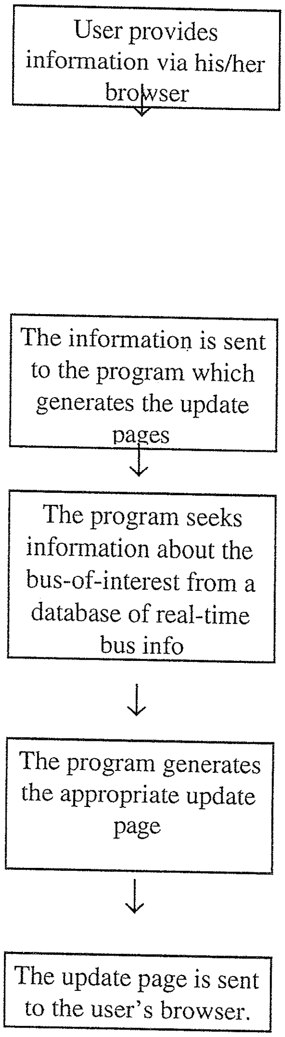
The TTIA1 program consists of an http server, a shell script, a Java program and a database. The http server is a large, fancy program and was available for use as a component of TTIA1 without needing modification. Though TTIA1 can't function without the http server, it's important to note that the server, unlike the other components, exists on the host computer separate from TTIA1 and is employed for many other purposes.

The shell script is a two-line program which simply starts the Java program. The Java program does the calculations for TTIA1, determining which message to display to the user and assembling the correct text. The Java program consults the database in order to do its work. The database is the only part of the four parts of TTIA1 which is not a program - it is a file.

Description of the System - Data Flow

As pictured below, the flow of data through the TTIA1 system is from a browser to the program which generates the update message and back to the browser.

Simplified:



User provides information via his/her browser

The information is sent to the program which generates the update pages

The program seeks information about the bus-of-interest from a database of real-time bus info

The program generates the appropriate update page

The update page is sent to the user's browser.

Detailed:

The user-provided information received by the http server is sent to TTIA1shell by means of the Common Gateway Interface (cgi-bin) facility of the server.

The information received by TTIA1shell is sent to TTIA1.class by means of Unix environment variables and standard in/out.

The information needed by TTIA1.class from TTIA1.data is fetched using Java's I/O facilities.

Finally, the data flows back to TTIA1shell via Java's I/O facilities, from TTIA1shell to the http server via the cgi-bin facility and standard out.

TTIA1 and Other Recent Research

TTIA1 is part of the general category Intelligent Transportation Systems (ITS) research. Among the goals of ITS research is the use of information, processing power and communications devices to improve public transit systems. The cutting edge of such research involves new systems which make vast quantities of data available. For example, Automatic Vehicle Location (AVL) systems provide real-time data about the location of buses. This kind of information is so new to ITS research that few applications exist for using the information.

The motivation for the development of TTIA1 was the installation of an AVL system on some Tri-Met buses. That AVL system is designed to help bus dispatchers, but has, as a secondary function, the ability to send information about bus schedule deviations to other applications. This information will be used by the customer service telephone information service (ATIS)[1] to provide real-time information as part of trip planning. The AVL data is also used for retrospective analyses of operations. Some plans exist for using AVL data to provide real-time information on a display at some bus stops and to other customer information systems.

As TTIA1 was under development before the AVL equipment was installed, it does not have access to the AVL data in real-time. This is one of the reasons that TTIA1 is a prototype of a messaging system. A later version of TTIA will need to manage the AVL data in a manner more sophisticated than TTIA1. TTIA1 simply uses a file which holds information that represents a snapshot of a the real-time AVL data. The managing of the messages from the AVL system is a challenge faced by both the trip planning implementers and TTIA implementers.

Many public transit providers in the U. S. are in the process of planning and/or developing systems like TTIA1. Only one system for providing real-time data to riders was operational as of the summer of 1995[i]. That system, Travlink, operating in Minneapolis, displays status reports for buses at some bus stops. The displays are not unlike the ones often seen at airports - they identify the bus and give the scheduled and actual arrival times. The software that performs this task is available to customers who request it.

Since systems like Travlink and TTIA1 are just emerging, the effects on riders and ridership of real-time information is not known. Research in this field will be active in the next few years. Beside the obvious questions of rider convenience and satisfaction, the question of the reputation of transit provider needs to be asked. Will the provision of real-time information bring too much attention to the problems of service? Also, researchers will need to weigh the financial costs and benefits of these systems.

[1] Automated Travel Information System, a product of Tidewater Consultants Inc.

TTIA1 is in the main stream of research in this area. AVL systems are likely to become more sophisticated and more reliable inspiring the development of many applications for disseminating real-time information.

Issues and Alternatives

Planning and implementing a prototype messaging system has illuminated some significant issues for the design of an operational messaging system. We present, in this section, some of the problems we encountered and solutions we chose as TTIA1 was constructed.

As mentioned above, the technologies of the World Wide Web were chosen as the user interface for TTIA1. For prototyping, this has the advantage that good off-the-shelf software is available both for browsers and servers. But this interface requires the user to go through several steps to use TTIA1. A production TTIA1-like system might send messages to the users' screens without requiring them to take any action. This would allow the program to be a reminder system - telling users when to leave to walk to the bus stop.

The implementation challenges of such a reminder system are these: software must be distributed to the user; the accuracy of the user's local clock must be confirmed; configuration of the program locally must allow the user to enter their walking time as well as all the information TTIA1 currently collects; with each access to the AVL data, the version number of the user's software must be checked. The major implementation issue for this kind of messaging/reminder system is automating the process of querying the database of AVL information. This would require the software obtained and installed by the user to send a request on the Internet instead of relying on a user clicking to send the request.

The software that processes the request from the user presented several implementation challenges worth noting. First, it is necessary to check each incoming request to see if it is reasonable. Unreasonable requests are those which are made long before the bus-of-interest could have sent pertinent information. As a rule, if the time at which the request is made precedes the scheduled time-of-interest by 45 minutes or more, no useful information will be available. A typical small schedule deviation (say ten minutes late) doesn't serve to predict the schedule deviation that will exist 45 minutes later. A bus might easily get back on schedule. A large schedule deviation (say _ hour late) is also not useful information 45 minutes ahead of time. A bus running very late may be passed up by the bus scheduled to follow it. Or another bus may be dispatched to replace the very late bus. Consequently, the time-of-interest and the time at which the request was made need to be compared. A message stating that there is no information about the bus at this time is required to avoid the problem of mindlessly reporting the current schedule deviation as if it were significant when it is not.

A second concern about interpreting the schedule deviation information before communicating with the TTIA1 users is broken-down buses. If the bus reports being five minutes late, two minutes later reports being seven minutes late and continues not to move, it's desirable to tell the user something more than the raw schedule deviation datum. If the AVL system at Tri-Met included a 'bus broken down' message, this task would be greatly simplified.

Even the supplemental messages that are a part of the AVL system present implementation challenges. Some text messages are provided by the system. Text messages such as "mechanical difficulty" could be made available to users of TTIA1. But other text messages may communicate information that a transit provider doesn't care to share with riders. It was decided that the TTIA1 implementation will disregard all messages from the buses except the schedule exception messages. This and the above mentioned decisions about the part of TTIA1 that assembles the appropriate update screen might need to be reconsidered when planning a production TTIA system.

Other decisions that affected the design of the prototype TTIA system concern the web pages. Having decided upon the appearance of the pages, there were significant challenges in the web page construction. These concern the number of web pages, the tools for building them, the data they are made from and the schedule for changing them.

The number of web pages needed for the prototype is determined by the number of bus routes involved. For each of the ten bus routes we used in our implementation (about ten percent of the bus routes of Tri-Met), there is one web page offering the day/direction choice. These ten pages branch to six pages each. The six branch to 10 or 12 pages each. These pages have in them the links to the Java program. There were 30,000 such links to construct. Each of these is a unique combination of a route number, a day of the week, a direction of travel, a bus stop and a time-of-day. The relation which specifies the 30,000 links has a tidy mathematical expression:

$$\langle r, d1, d2, s, t \rangle$$

where *r* is a route number

d1 (*d* for day) is one of three choices: weekdays, Saturdays, Sundays and holidays

d2 is one of two directions corresponding, roughly, to inbound or outbound travel

s is a bus stop

t is one of the times-of-day that the bus corresponding to $\langle r, d1, d2 \rangle$ is scheduled to stop at stop *s*

Since each of the 30,000 links is a URL which starts the Java program and returns a web page to the user, it is possible to create 30,000 web pages instead of 30,000 links. In fact, every web page that is part of the TTIA1 system could be produced by a program rather than being written to a file which a browser can access. Thus it was necessary to decide which of the TTIA1 web pages ought to reside on the computer as files and which should be produced as they are needed. The division used for TTIA1 - all web pages are stored in files except the update report pages - reflects the division of the information TTIA1 users need. Some information, for example the list of the (ten) routes which are part of TTIA1, doesn't change. This information lends itself to a web page that is stored in a file. But other information, most obviously the schedule deviation of a bus, changes often. Thus it lends itself to a dynamically produced web page. This same division can be seen in the existing Tri-Met web pages. Since real-time information is not currently included, all the pages are stored in files. This has the tremendous advantage of conceptual simplicity: a web page is just a file that a browser can look at.

The production of the static web pages (those stored in files) deserves a few words. The ten routes offered by TTIA1 require about 600 web pages. This is too many to produce by hand. Perl, a programming language which is good at file manipulation, was used to modify the existing Tri-Met web pages which produces the TTIA1 pages. Using web pages as the input to a program that produces web pages was a poor choice. The Tri-Met web pages are produced by a program that has information from the Tri-Met database as its input. We were wishing to avoid repeating the work of transforming database information into web pages. However, there is one small piece of data that TTIA1 needs which the Tri-Met web pages does not include (it is the train number for each of the 30,000 links). So database information was needed for the production of the TTIA1 pages after all. It is possible that using web pages as input to the Perl program made the web page production more complex.

The production of the static web pages brings up another issue - how static are the static pages? If a new version of TTIA accommodates additional routes, the pages will need to change. Also, the bus schedules change and the TTIA1 pages will need to reflect those changes. If the TTIA1 pages are created using Tri-Met's web pages as input, then the TTIA1 pages should change when the Tri-Met pages are changed. Exactly when will we change the TTIA1 pages? This question affects the decision about the appropriate input to the program that produces the TTIA1 pages.

Finally, the best balance between static web pages and dynamically produced web pages is influenced by the use of the TTIA1 system. If we expect 100 users each day, most of the static pages will never be examined and it is probable that no two users will ask for the same update report page. But 10,000 users might cause the update report page for some specific bus to be produced by the Java program over and over again. Heavy use of a page would encourage the implementer to store that page in a file - even if the file needed to be updated every few minutes. For the usage we expect, update report pages don't need to be cached.

These implementation issues and alternatives were the major influence on the design of TTIA1. Some decisions seem to have been justified and constitute good advice for the design of any TTIA1-like system. Others were unwise and need to be changed in a new version of TTIA. As TTIA1 was built, implementation challenges have been tempered with input from potential users. Users' suggestions have introduced several changes to TTIA1 and will be a part of the implementation decisions of the future.

User Input

About a dozen potential users of a commercial TTIA system were interviewed concerning their opinion of the prototype, TTIA1. Three kinds of comments were made about TTIA1; they concern appearance, functionality and value.

The comments made by potential users about TTIA1's appearance included a request for predicted arrival times to appear in a different color if the bus is not on schedule, a request for a format that matches the displays in an airport, and a request for bus stops to have numbers rather than names. Some less significant comments pointed up spelling mistakes and poor choices for html formatting. Specifically, the page on which the user chooses their time-of-interest for the bus they have already chosen has a list that is so long that it goes off the screen. A table or a scrolling box would be better. At first glance, several people thought the update page showed the previous and next bus rather than the scheduled and updated times.

The functioning of TTIA1 would be improved by adding the current time to the update report, one person said. Also, there was a request for the program to compute the time at which a person should leave his or her desk to walk to the bus stop. The ability to predict the bus' likelihood of making a certain connection was mentioned several times as a necessary part of a useful TTIA system. An extension to the links available from the update report page should be made to include a 'query again' button, a 'back to the schedule' button and a 'back to all routes' button. Reroute information, especially for snow routes, and stop closures were mentioned as significant for a TTIA user.

The people who looked at TTIA1 had varying opinions of the usefulness of the system (even with its ideal functionality). Nearly every user suggested that this information would be useful at the bus stop. For example, one person commented that he would like to know, when he arrives at the stop, whether the bus has already gone by. For riders whose buses come by only twice an hour, this would allow other tasks to be accomplished in place of waiting at the bus stop. In response to the plan to incorporate an automatic reminder into TTIA1, one user objected to his computer doing anything that he had not initiated. He did not want to be bothered with windows that pop-up. Finally, most potential users interviewed said that they would use a TTIA-like system daily. One said she would not use it unless she had an appointment to keep - for normal travel she is not concerned about a little extra time waiting at a bus stop. Another person reported that he would use a TTIA-like system in his capacity as the Tri-Met coordinator at his place-of-business. One person who did not know of TTIA1 recently wrote email to Tri-Met asking for a TTIA-like service.

The input from users which resulted in immediate changes to TTIA1 is not mentioned here, since the effects can be seen in the Web pages above. Our thanks to all the potential users of a TTIA-like system who looked at TTIA1 for us.

Strategy for Prototype Development

For those interested in developing a TTIA-like program, a variety of software engineering approaches are available. One possibility is rapid prototyping. Another is top-down development. The following is a suggestion for incremental development of a TTIA-like program which uses some ideas from the rapid prototyping approach and some from the top-down approach.

1. Write an html document that offers schedule update information for one route. Create links which purport to give real-time information but actually always return a "no information at this time" page.
2. Build a cgi-bin program that takes no arguments (or ignores its arguments) and returns a "no info at this time" page. Change the links so that they call this program but send no arguments.
3. Produce a full set pages that guide users to their bus-of-interest. Have each bus of interest link call the cgi-bin program.
4. Enhance the full set of pages so that the links send appropriate arguments (route number, stop, time-of-day, etc.) to the cgi-bin program which ignores its arguments.
5. Enhance the cgi-bin program so that it reports the arguments it received but still claims no information is available for the bus-of-interest.
6. Enhance the cgi-bin program so that it uses its arguments to search the database of schedule exception messages for information about the bus of interest and reports that information was found.
7. Add to the cgi-bin program algorithms to interpret the schedule exception information and send an appropriate message to the user.

Future Directions

The most important feature of a new and improved TTIA (TTIA2) will be the ability to make use of the actual schedule exception messages. The data used by TTIA1 was, as mentioned above, created by hand to resemble actual data from the AVL system. Real data will be available in the Fall of '96, enabling work on this component of TTIA2.

Another important but not profound part of a TTIA2 system will be a better method of producing web pages. The form of the input data is an issue,. This was also considered above.

Future work may enable reporting of reroute information. Since text messages are available to the customer information operators giving reroute information, in may be possible to use them for TTIA2 also. Rerouting may be planned or may emerge. These produce different kinds of reroute information and require different reporting to customers. A person on the phone has a significant advantage over a computer in giving reroute information.

A simpler and very useful extension to TTIA1 would be the ability to report whether a bus is on its snow route or not. The experience of the customer information operators suggests that use of a TTIA-like system would be heavier if the snow route information were available.

More ambitious are the plans to include maps in a future version of TTIA. In addition to allowing users to access information in another mode, we hope to aid users who don't know the location of their stop by its street intersection name. Even more help may be made available for users trying to locate their bus stop. A more complete list of stops and an interface for choosing from the list may include reference to the map of that area.

Distribution of a future version of TTIA would allow the implementation of a reminder system. If TTIA were resident on a users home or office computer, a message could be sent to their screen at a time they choose telling them that it's time to go to the bus. Should the bus be running late, this information would be communicated also. In a highly configurable system, a user could ask for information on some alternative bus when the first bus is late.

Conclusion

The production of the TTIA1 system has demonstrated of the feasibility of a real-time bus schedule update program for the Web. A second significant result is the collection of issues and alternatives that sum up what we've learned by building TTIA1. By implementing one more version of the TTIA prototype we hope to learn more about the best data to use for user interface construction, the userpreferred format for the update messages and the possibilities of a reminder system. ITS systems providing similar information to customers should be increasingly available for comparison, allowing TTIA to contribute to the development of new, sophisticate information delivery systems for transit.

[i] Review and Assessment of En-Route Transit Information Systems, U. S. Department of Transportation, July 1995