Editor: Alberto Broggi
University of Pavia, Italy
broggi@ce.unipr.it

MyBus: Helping Bus Riders Make Informed Decisions

Stuart D. Maclean and Daniel J. Dailey, *University of Washington*

Many intelligent transportation systems will heavily influence our way of living in the future; some of them are in early design, while some are already under test. This installment describes a system already operational in the Seattle area; it lets users access information about, and forecasts of, bus schedules. The full availability of such systems on a large scale, taking into consideration not only buses but also trains and planes, will allow efficient planning of people's mobility.

Such large-scale systems will allow rerouting of transportation systems or manipulation of traffic lights to intervene in real time in traffic jams or unexpected delays to smooth the traffic flow. Furthermore, the availability of a sufficiently complete statistical analysis of driving times will permit replanning of bus schedules and routes, on the basis of people's needs and utilization.

I welcome contributions on the current status of ITS projects worldwide as well as ideas and trends on future transportation systems. Please send your contribution to: broggi@ce.unipr.it; www.ce.unipr.it/broggi.

—Alberto Broggi

The Smart Trek Model Deployment Initiative, a US Department of Transportation-funded Intelligent Transportation Systems program in the Puget Sound region, has made great strides in integrating and disseminating traveler information. The initiative focuses on real-time information to help travelers make informed decisions about their travel options. The Smart Trek project has brought about a variety of real-time transit information applications. One of these is MyBus, which makes departure predictions and delivers traveler information to Web browsers and cell phones.

MyBus aims to present to riders, in real time, the predicted departure times of buses at specific locations throughout a transit region. King County Metro, Seattle's transit agency, operates a large fleet. Up to 1,200 vehicles are in service simultaneously, departing from over 1,000 locations. MyBus predicts approximately 210,000 weekday and 140,000 weekend scheduled departure events. This is over a million departure predictions per week. MyBus has shown that predictions on this scale are feasible and manageable.

Several technologies are central to the success of MyBus. A common format for the transit agency schedule and spatial data was crucial, letting us redeploy the Seattle pilot project with data from the Portland Tri-Met transit agency with minimal effort. The format we chose was a database schema based directly on the Transit Communications Interface Profiles standards.[1]

MyBus

MyBus is a distributed application (see Figure 1). Schedule data and real-time *automatic vehicle location* data streams flow between components. These components include a legacy AVL system as the source of real-time data, a prediction generator (the *predictor*), and a Web server for final text formatting and delivery over the Web. The extensive use of the ITS Self-Describing Data (SDD) protocol[2] for intercomponent communication ensures reliable and efficient data manipulation and transport. Reusable collaborative components simplify building new applications.[3] For example, the predictor's output goes to the Mybus Web server, but other components could also access this data. These downstream components could do such things as analyze on-time performance or estimate congestion.

Prediction

The predictor uses three inputs: a schedule data set, a set of historical trip realizations, and a real-time AVL stream. The schedule data provides each event's expected time, as well as its location along the *pattern* or path that the vehicle traverses. Historical data provides ensemble-averaged statistics to the algorithm that predicts vehicle departure. The AVL stream supplies instantaneous bus location information approximately every one to three minutes per vehicle. The latter two inputs are the most important; you can build a successful predictor without temporal schedule information.

1094-7167/01/$10.00 © 2001 IEEE

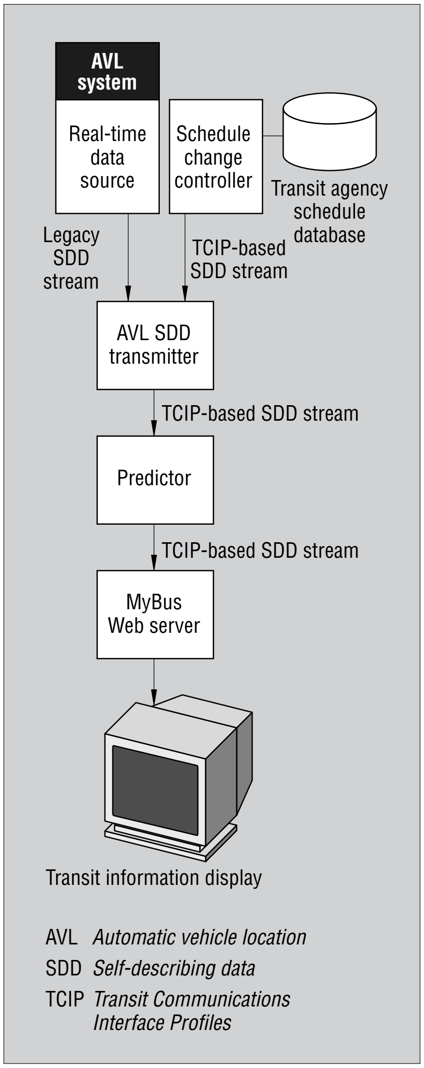


Figure 1. The MyBus architecture.

According to the TCIP standard nomenclature, a bus travels along a *block* (not to be confused with a city block), which comprises a sequence of *trips*. Each trip is a single run of a bus route in one direction—for example, the 10:30 a.m. Route 43 bus from Downtown Seattle to the University of Washington. Along each trip, scheduled locations called *timepoints* exist, generally at major road intersections. We designate each timepoint as a prediction goal. The prediction algorithm[4] tracks each bus to its many goals along the block.

The prediction algorithm uses an optimal filtering technique based on Kalman filter technology.[5] An underlying assumption is that any real-time arrival prediction algorithm depends on reliable real-time transit vehicle location information. When the maximum number of vehicles are in service (for example, approaching morning rush hour), the Seattle-based predictor produces up to 25,000 predictions every 10 seconds. To minimize statistical errors, MyBus starts a prediction process for each goal. It bases this process on the time before the scheduled departure or the distance from the goal. For example, the tracking of a bus could begin 30 minutes before the scheduled departure time or 20 miles from the goal. At each time step of 10 seconds, the predictor makes a new prediction for both the distance to the goal and the time until departure from the goal. Interleaved with this *propagation phase* are the real-time AVL inputs, which constitute the *update phase*. The state vector for each predictor is output at both the propagation and update phases.

In principle, MyBus can track a bus from the start of its block to any goal on the block. In practice, the schedule contains *deadheads*—out-of-service segments for each block. The current prediction process will not predict across a deadhead trip. That is, a goal appearing in a block just after a deadhead trip will not be associated with an active prediction process until MyBus has observed through the AVL updates that the bus has completed the deadhead portion of the trip.

The prediction algorithm produces the optimal estimate of the departure time given the information provided to the filter. However, predicting the future is always a challenge. To compare the predictions with the vehicle behavior, we need to estimate the actual departure time. Because we are tracking the vehicle irregularly, we have no guarantee that the location will be reported just as the vehicle departs. To estimate the real departure, we can record the location report just before the arrival and just after the departure and linearly interpolate the actual departure time. The filter continuously predicts the departure as a function of both space and time.

We can express the deviation of the predictions from the actual behavior as a probability surface in space and time. Figure 2a shows the probability of this deviation and the time until the departure for which MyBus made the prediction. Figure 2b shows these measures for the transit agency schedule. We created the surfaces in Figure 2 using the predictions made over the course of one

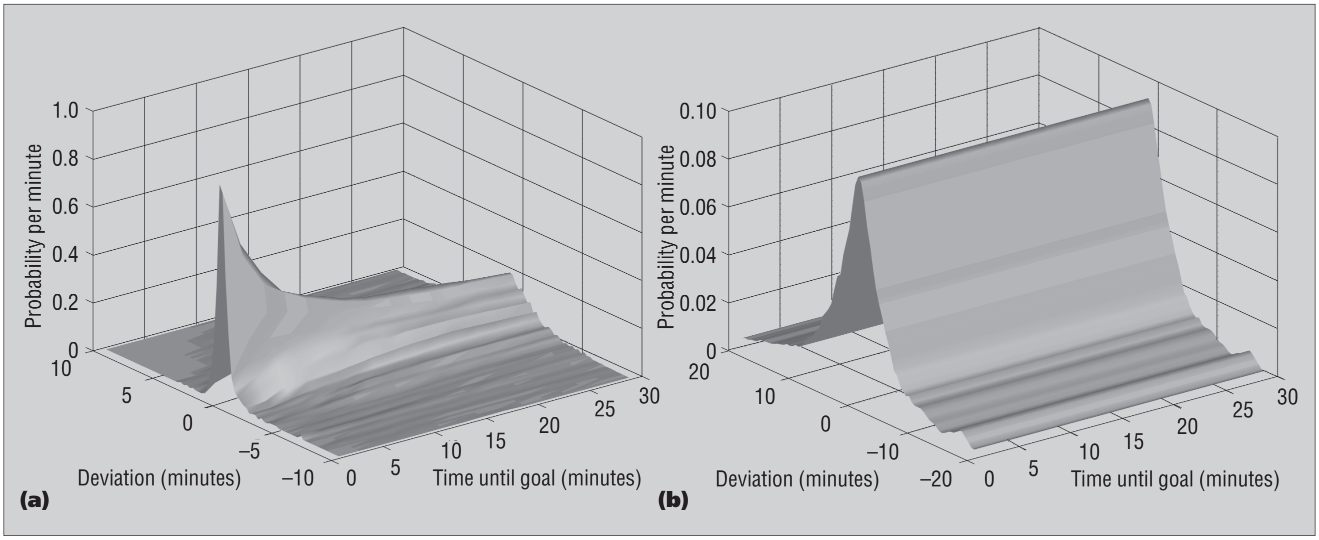
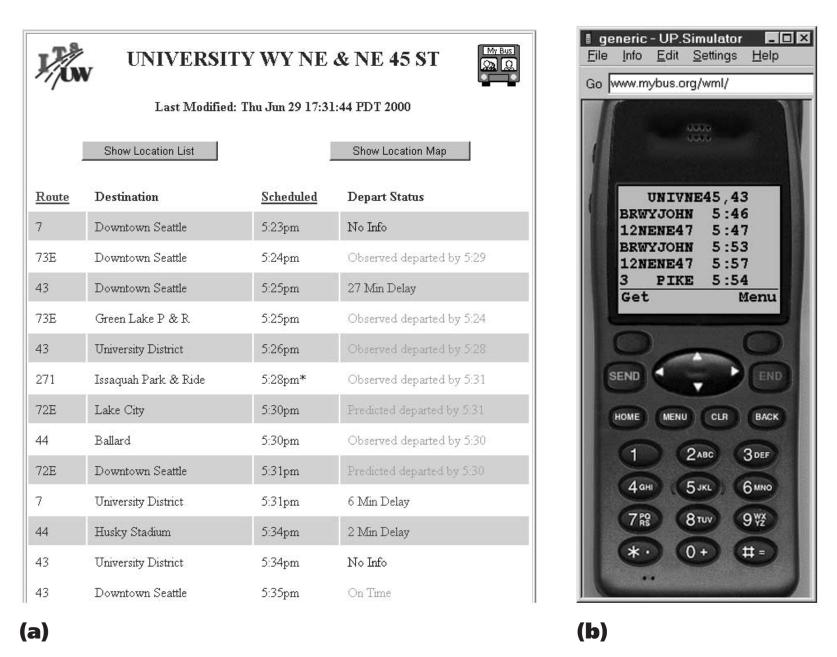


Figure 2. Deviation of predictions from the actual behavior for (a) MyBus and (b) the transit agency schedule.



Last Modified: Thu Jun 29 17:31:44 PDT 2000

Route	Destination	Scheduled	Depart Status
7	Downtown Seattle	5:23pm	No Info
73E	Downtown Seattle	5:24pm	Observed departed by 5:29
43	Downtown Seattle	5:25pm	27 Min Delay
73E	Green Lake P & R	5:25pm	Observed departed by 5:24
43	University District	5:26pm	Observed departed by 5:28
271	Issaquah Park & Ride	5:28pm*	Observed departed by 5:31
72E	Lake City	5:30pm	Predicted departed by 5:31
44	Ballard	5:30pm	Observed departed by 5:30
72E	Downtown Seattle	5:31pm	Predicted departed by 5:30
7	University District	5:31pm	6 Min Delay
44	Husky Stadium	5:34pm	2 Min Delay
43	University District	5:34pm	No Info
43	Downtown Seattle	5:35pm	On Time

```
  UNIVNE45,43
BRWYJOHN  5:46
12NENE47  5:47
BRWYJOHN  5:53
12NENE47  5:57
3   PIKE  5:54
Get       Menu
```

Figure 3. Delivering MyBus information: (a) the MyBus Web Page and (b) a Wireless Application Protocol (WAP) phone emulator.

day for the second-busiest location in Seattle Metro Transit's service. Comparing these two surfaces suggests that prediction errors from dynamic information are 50 to 75 percent smaller than those from using the schedule alone. Moreover, predictions are accurate both long before the scheduled departure time and near departure time. Rider access to such information is vital in encouraging mass transit.

Presentation

The Web server receives input from the predictor and stores and formats this data for output over the Internet. To deliver the final prediction information, MyBus uses the standard HTTP/HTML or HTTP/WML (Wireless Markup Language) combinations. This assumes that the delivery site has an HTML/WML interpreter, which in turn implies some level of computing capabilities at that site. The most common form of delivery is a browser-like screen or a Wireless Application Protocol (WAP) phone that can display WML (see Figure 3).

MyBus uses a set of prediction states to determine the message sent to the client. Two *departed states* exist. The first, *bus-observed-to-have-departed-goal*, indicates that MyBus has received an AVL update that shows a bus past the goal. The second, *bus-predicted-departed*, indicates that, in the absence of data, the Kalman filter has predicted that the bus has departed. Formatted messages directly reflect these states (for example, see Figure 3a, rows 6 and 7). MyBus also provides timeliness reports such as "On Time" or "15 Min Delay."

The Kalman filter produces a covariance that measures the prediction's validity. When a vehicle has not reported its location for a long time, this measure becomes large, and eventually the prediction reaches an unacceptable level of significance. In such cases, MyBus switches the display message to "No Info."

Any number of rule sets for mapping predictions into displayed messages are possible. Because the predictor and Web server are distributed, we can easily substitute a new display application.

Implementation

The predictor, Web server, and supporting SDD applications are all built in Java.[6] Currently, the Seattle predictor and the combined Seattle/Portland Web site run as two Java Virtual Machines on the same Windows NT host. The ratio of dynamic to static content available from the Web site is high. This requires a mechanism for producing the dynamic content and interfacing it with standard Web server software, such as Apache (www.apache.org). To this end, MyBus uses Java servlet and Java Server Pages technology.[7]

The SDD component library used for data transport within the MyBus hierarchy is available at www.its.washington.edu/

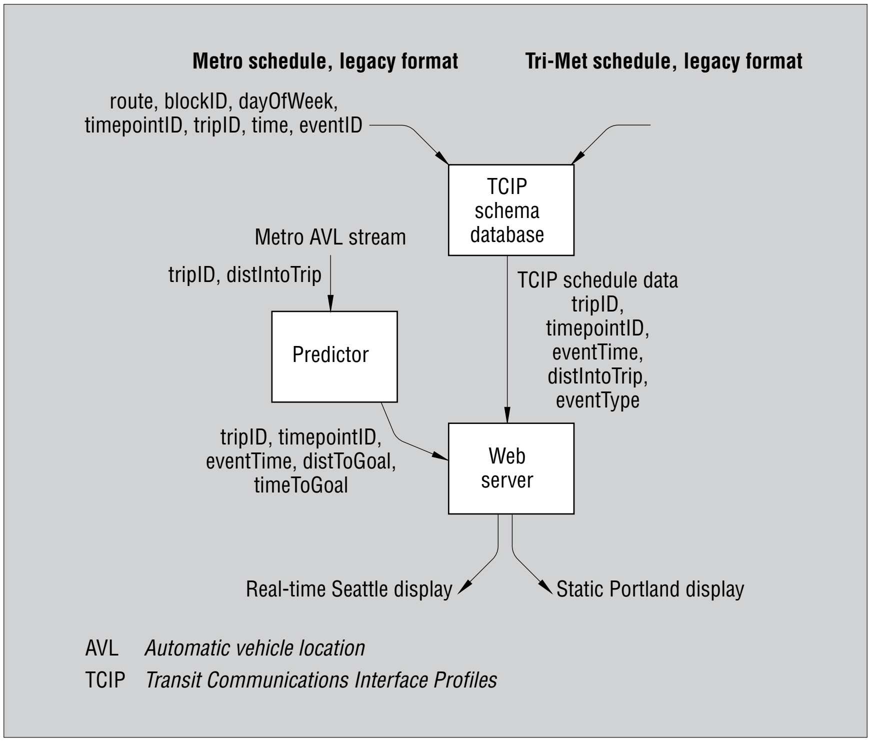


Figure 4. A MyBus system diagram, showing data conversion from a legacy format to the standard TCIP schema, with real-time and static prediction streams.

bbone. You can try MyBus at http://mybus.org.

TCIP and MyBus

The TCIP is a set of interface standards for the US transit industry. The domain covers the data needs of the functions related to the support of public transportation operations, service, and planning.[1] This includes input and output data for scheduling and passenger information, which are of particular interest to MyBus. The TCIP's crux is the definition of data elements and how they are presented.[1]

Pertinent standards

From a software architecture standpoint, TCIP is a benchmark for data structure names, types, and relationships. For a rider-directed information system such as MyBus, three standards are of interest:

- the Standard on Passenger Information Objects (NTCIP 1403),
- the Standard on Scheduling/Run-Cutting Objects (NTCIP 1404), and
- the Standard on Spatial Representation Objects (NTCIP 1405).

All are available for review at www.ite.org/standards/ntcip/index.htm.

Integrating TCIP

The schedule information from the target transit agencies can be represented in terms of TCIP data-dictionary data elements with the additional relational keys necessary to couple the data elements. Each agency is likely to manage its schedule data differently, so each agency's data must be transformed into the TCIP format. This transformation makes the data available to MyBus. For example, we've successfully mapped the data elements from both King County Metro and Portland Tri-Met into this framework. This mapping has let us construct a TCIP-based database.

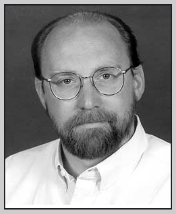
At the simplest level, we map TCIP data elements—for example, timepoint, pattern, and trip—into an SQL schema. We call this the standard TCIP schema. We use the schema's tabular structure to create the components that support the MyBus application interface. Both spatial and temporal schedule information are necessary to support any predictive algorithm, and both contribute to the schema; for example, a timepoint table and a trip table both exist. The combination of a real-time AVL stream with TCIP schedule information lets MyBus make real-time predictions on bus departures. Without the AVL component, MyBus can use the TCIP schedule data to produce a scheduled or "timetable" interface. Such a system will typically present richer content than available through an agency's published schedules. Figure 4 demonstrates these two possibilities that both rely on the availability of TCIP schedule data in the context of Seattle and Portland.

MyBus combines optimal estimation, distributed computing, information technology, and the World Wide Web to produce an Advanced Public Transportation System information system. It supports existing Web browsers as well as the latest handheld devices. Because MyBus conforms to the TCIP standard, it is portable to any US transit property. It is constructed in a portable language, so it can operate on a variety of computing platforms. Its modular construction allows for future interfaces. Interfaces under design include voice synthesis and recognition systems and interfaces to other advanced-traffic-management systems where the transit vehicles act as probes to measure traffic congestion.

References

1. *Transit Communications Interface Profiles (TCIP) Framework Document,* Draft 97.01.02, Joint AASHTO/ITE/NEMA Standards Publication NTCIP 1400, Inst. of Transportation Engineers, Washington, D.C., 1999; www.ite.org/standards/ntcip/1400v0102.pdf (current 26 Jan. 2001).
2. D.J. Dailey, M.P. Haselkorn, and D. Meyers, "A Structured Approach to Developing Real-Time, Distributed Network Applications for ITS Deployment," *Intelligent Transportation Systems J.*, vol. 3, no. 3, 1996, pp. 163–80.
3. D.J. Dailey, D. Meyers, and N. Friedland, "A Self Describing Data Transfer Methodology for ITS Applications," *Transportation Research Record*, no. 1660, 1999, pp. 140–147.
4. D.J. Dailey et al., "An Algorithm and Implementation to Predict the Arrival of Transit Vehicles," *Proc. IEEE Intelligent Transportation Systems Conf.*, IEEE Press, Piscataway, N.J., 2000, pp. 161–166.
5. B.D.O. Anderson and J.B. Moore, *Optimal Filtering*, Prentice Hall, Upper Saddle River, N.J., 1979.
6. K. Arnold and J.D. Gosling, *The Java Programming Language*, Addison-Wesley, Reading, Mass., 1996.
7. J. Hunter and W. Crawford, *Java Servlet Programming*, O'Reilly and Associates, Sebastopol, Calif., 1998.

Stuart D. Maclean is a research associate at the Intelligent Transportation Systems Research Project at the University of Washington's College of Engineering. His research interests are programming models and environments, particularly for parallel and distributed programming; visual and graphical programming; object-oriented programming; software architectures and software engineering; programming language design and implementation; and real-time systems and the application of object orientation to such systems. He received his BSc in mathematics, his MSc in operational research, and his PhD in computer science, all from the University of Southampton. Contact him at Intelligent Transportation Systems, College of Engineering, Univ. of Washington, PO Box 352500, Seattle, WA; stuart@its.washington.edu; www.its.washington.edu/~stuart.



Daniel J. Dailey is an associate professor in the University of Washington's Department of Electrical Engineering and the director of the department's Intelligent Transportation Systems Research Program. His research interests are intelligent transportation systems, quantitative methods for modeling of physical phenomena, optimization, and distributed computing and communications applied to information technology. He received his BS in nuclear engineering from Pennsylvania State University and his MS and PhD in nuclear engineering from the University of Washington. Contact him at Intelligent Transportation Systems, M322 EE/CSE, Box 352500, Univ. of Washington, Seattle, WA 98195; dan@its.washington.edu; www.ee.washington.edu/people/faculty/dailey.