

**Paper No. 970397**

**PREPRINT**

**Duplication of this preprint for publication or sale  
is strictly prohibited without prior written permission  
of the Transportation Research Board.**

**Title:** Seattle Smart Traveler

**Author:** D. J. Dailey, D. Loseff, D. Meyers, and M. P. Haselkorn,  
University of Washington, Department of Electrical Engineering,  
Department of Technical Communication, Seattle, WA

**Transportation Research Board  
76th Annual Meeting  
January 12-16, 1997  
Washington, D.C.**

# Seattle Smart Traveler

D. J. Dailey, D. Loseff, D. Meyers, and M. P. Haselkorn

University of Washington  
Department of Electrical Engineering  
Department of Technical Communication  
Seattle, WA

## ABSTRACT

Seattle Smart Traveler (SST) is a world wide web (WWW) application designed to test the concept of “dynamic” rideshare matching. SST collects spatial and temporal trip information using a series of WWW pages, performs a match using SQL specifications to a database engine, and supports both the standard phone-based contact methodology as well as two new, unique email-based contact methodologies. (<http://sst.its.washington.edu/sst> )

**KEYWORDS:** carpool, dynamic ridematch, rideshare, world wide web, database

## INTRODUCTION

Seattle Smart Traveler (SST) is a Federal Highway Administration field operational test (FOT) which is presently underway. SST is designed to test the concept of “dynamic” rideshare matching. We define dynamic ridesharing as “the sharing of a single trip by two or more individuals, without regard to any previous history among the individuals involved.” (This is similar to the definition of real-time ridesharing in [2].) Additionally, SST defines a “trip” as “a single instance of travel from one geographic location to another.” A “trip” is, therefore, by definition, a one-way trip, and the popular conception of a “round trip” is, for the purposes of SST, two “trips.” This is an important distinction between “dynamic” and “traditional” ridematching systems.

Dynamic rideshare matching differs from traditional rideshare matching in two important ways. The first major difference is the treatment of the traveler's schedule. Traditional systems assume the traveler has a fixed schedule and a fixed set of origins and destinations. [5] A dynamic system must consider each trip individually and be able to accommodate trips to arbitrary points at arbitrary times by matching users' individual trips without regard to trip purpose. The second major difference is that dynamic ridematch systems must provide the match

information to the user quickly in order to accommodate near-term (same day) travel as well as long-term (future days or weeks) trips. Traditional systems frequently provide a matchlist through paper mail, a process which may take more than a day. [7, 10] For these two reasons, the requirements for dynamic rideshare matching are more demanding than those for the traditional rideshare application.

In designing SST, we postulated that the users of such a system view carpooling as a travel option in terms of three types of trips: (1) trips that are part of a commuting schedule, both the “traditional” fixed commuting schedules, as well as the regular but variable commuting schedules, (2) additional recurring trips, not part of commuting, and (3) the occasional trip or single trips to single destinations. Because the University environment inherently provides examples of all these types of trips, we chose to develop the SST application at the University of Washington (UW) using a world wide web (WWW) interface and to use members of the UW community as the testbed for the FOT. The UW is an example of a large, closed environment where potential users have multiple, highly variable schedules and a financial incentive to arrive at the campus without a car. Additionally, the UW is an example of an environment with a high level of technological sophistication, where most potential users are computer literate and have access to multiple communications technologies (e.g. email, voice mail, and paging), and such access has been identified as critical in the assessment of advanced rideshare technologies. [1, 3, 6] This combination of variable schedules, computer literacy, and access to multiple communications media makes this population a reasonable representation of future work environments for employees of large agencies and an ideal testbed for “dynamic” rideshare matching.

## **SST DESIGN**

SST is designed to be a complete rideshare matching system, capable of providing matches for those with traditional needs as well as for those with variable schedules and occasional needs. This design recognizes that the traditional ridematching requirements are only a subset of the requirements for a dynamic rideshare system.

SST is designed to be accessible through the World Wide Web. This design decision was driven by two features of the WWW: (1) the popularity of the WWW along with the wide availability of free browser software and (2) the WWW is available 24 hours a day, 7-days a week. These two features ensure that SST can be delivered to a wide audience at the individual user's discretion and convenience. These features were also deemed necessary to entice a sufficient number of users to reach the critical mass where dynamic ridematching actually takes place.

In this paper we present the design of SST in two parts. We first review the matching information that is collected and the spatial and temporal matching scheme employed in SST, and we then review the technical design of the SST program.

### ***Design: Collection of User Information***

SST is set up as a series of Hyper Text Markup Language (HTML) forms to collect required information from the user. The opening page of SST requests either (1) a new user's student or staff number or (2) an established user's userid and password. Both these actions force authentication of the user. SST is presently an experimental program which limits the user group to the University of Washington community, and the authentication process enforces this restriction. The SST application also requires contact information and a password. The requested contact information is a phone number and an email address. The email address is important because SST is able to automatically generate and send email messages to users with matching schedules. It is noteworthy that the user's home address is not requested as part of the SST registration. A home address is not used for matching purposes, and there is no paper copy of a matchlist to mail out, so a home address is not needed. The items just mentioned (email, phone number, and a password) are the total extent of the personal information collected as part of the SST registration process. To perform ridematching, mechanisms to collect and manage individual trip data are necessary, and the methodologies used by SST to accomplish this are described in the following section.

### ***Design: Matching Temporal/Spatial Domains and Contact Methodology***

Information collected by SST is always specific to the individual trips. From a user perspective, trips are divided into three categories in SST: (1) regular commute trips, (2) additional recurring trips (a trip made on a regular basis, in addition to a commute trip), and (3) an occasional trip. However, all three types of trips require essentially the same temporal and spatial information: (1) trip spatial origin, (2) trip spatial destination, (3) day of the week, (4) trip-departure time range, and (5) trip-arrival time range. In addition, all trips are assigned an expiration date. This expiration date is assigned automatically for the first two types of trips, and the user specifically selects the expiration date for an occasional trip. The expiration date is used to remove trips whose temporal relevance has expired, an important feature sometimes overlooked in ridematch systems that do not age the trip information.

In the SST system, two important design decisions were made regarding the collection of the temporal and spatial information required to perform ridematching. The first decision involves the collection of temporal data. SST requests that the user enter a range for both the departure and arrival times. This makes the level of flexibility in a trip schedule explicit to the user and under the control of the user entering the travel information.

The second design decision involves the spatial information for origin and destination. The spatial information is requested with a series of pull down menus

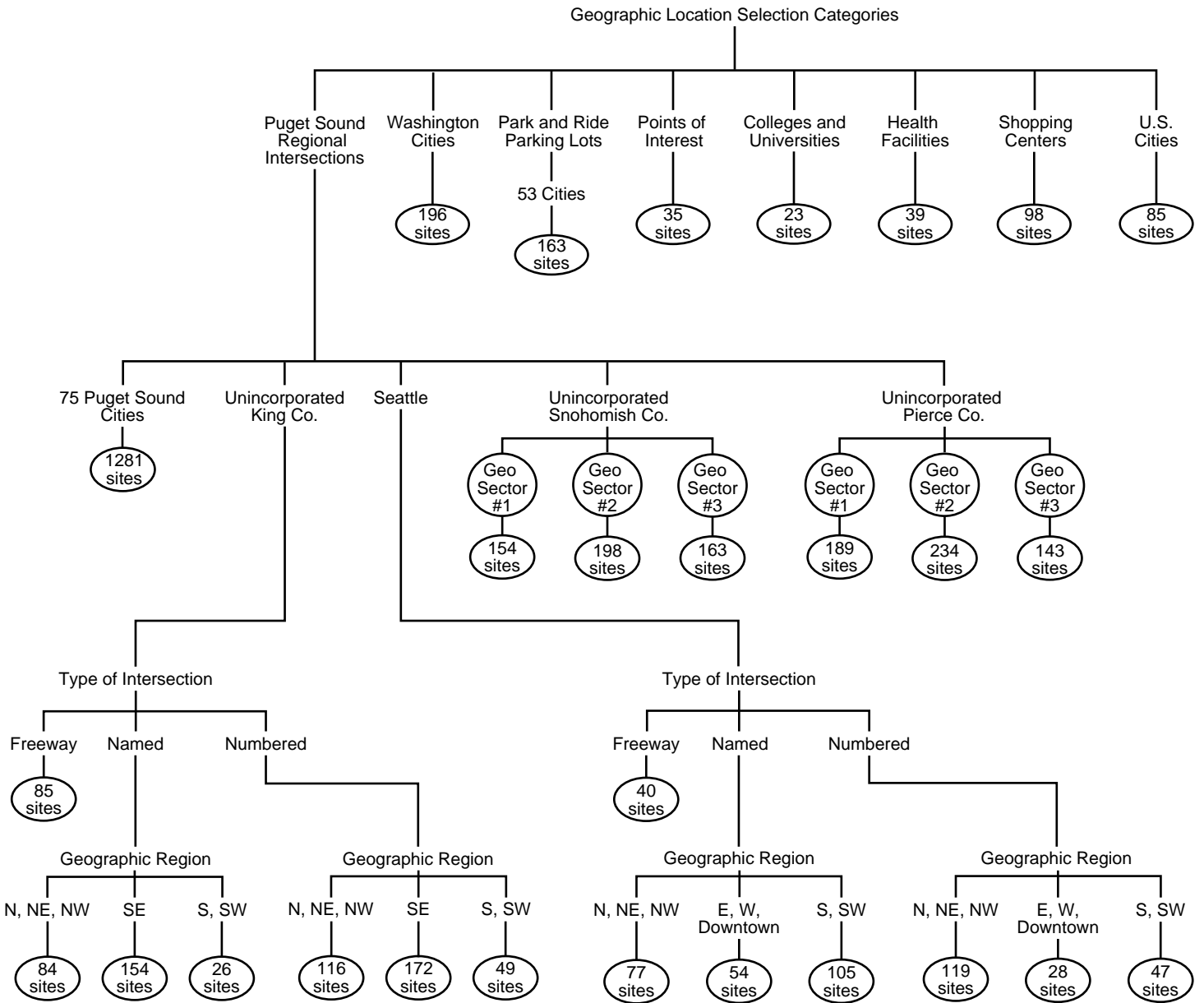


Figure 1: Geographic location search trees

that implement an efficient search tree. At the top of the tree, the user can select from eight landmark types. The tree structure for selecting spatial information is shown in Figure 1. This tree structure is used to sequentially reduce the area of the search until a single landmark can be identified. The depth to which the user must descend in each tree depends on the starting point. For example, to get to a specific road intersection in Seattle, the user must descend four levels (the greatest depth in the entire tree). The user starts at Puget Sound Intersections in Figure 1, proceeds to select the city of Seattle, decides that the street is either a freeway, a named street, or an intersection identified by numbered streets, and finally selects the specific intersection from a list. In contrast, to select many other spatial landmarks, only two choices need be made (see Shopping Centers in Figure 1). This reliance on landmarks to describe the trip endpoints removes the need for origin and destination addresses as well as the need for GIS software, but it does limit the total number of origins and destinations that are available to the user. (The implementation of SST uses latitude and longitude geo-coordinates of over 3,500 locations in the Puget Sound region, 200 cities in Washington State outside of the Puget Sound region, and 100 cities across the nation. The coordinates are taken from the TIGER spatial data database.)

Once temporal and geographic information is collected, the matching is done using a database engine and SQL commands. The database is queried for entries that match temporally and then geographically. The temporal match finds pairs of participants whose start and end point temporal ranges overlap (recall that the user explicitly enters ranges for the departure and arrival times). The default geographical matching range is 15% of the length of the trip in all directions from each endpoint to match against other user endpoints. This geographical coverage is user configurable and can be varied by broadening (25%) or narrowing (5%) the scope of the search. (It is noteworthy that while this matching approach is GIS based, it uses variably sized areas around end points rather than route-based matches as in [8].) Once a set of matching trips has been produced, the names, phone numbers, and email addresses of users with matching trips are displayed on the screen. Riders who have a matching trip schedule must contact each other, and SST takes a unique approach to supporting this contact.

SST provides a matchlist with phone numbers and email addresses; however, in addition to a simple matchlist, an automated email option is integrated into SST. This option allows a user who has found matches to have pre-formatted email automatically sent to one or more of the other users with matching trips. This message contains the sender's contact information and trip details. In addition to basic email, SST provides a functionality that permits the user to email a message to a remote user's Seiko message watch. The message contains either a phone number or a notification of the arrival of email. These two email contact methodologies, used to augment the more traditional match list, are unique to the SST program.

The SST program collects spatial and temporal trip information using a series of WWW pages as described, performs a match using SQL specifications to a database

engine, and supports both the standard phone-based contact methodology as well as two new, unique email-based contact methodologies. The next section presents the actual mechanics of creating a ridesharing application that requires state information about the user in a stateless WWW environment.

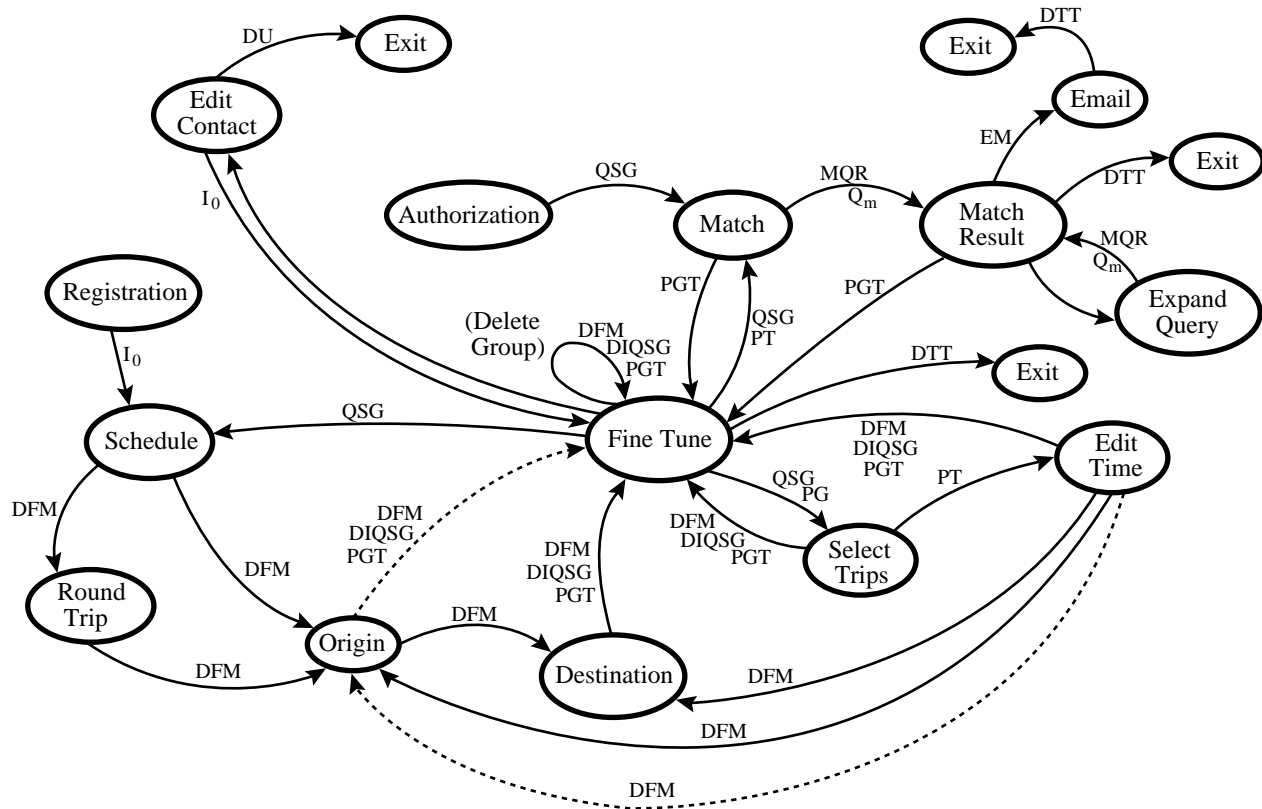
## **THE SST APPLICATION AND THE WWW**

In this section we describe the paradigms used inside the SST application. SST as a WWW application must address many of the problems that all program developers face in this environment. The main WWW problem encountered in developing SST is that browser interactions with a WWW server are inherently stateless. SST needs to maintain the state of an ongoing transaction (e.g. the user information about trips and contacts), and maintenance of that state information is difficult. In addition to the difficulty with state, most web browser applications have a facility that enables the user to return to a previously displayed page without notifying the server of that action in any way (the back button). The existence of a back button makes it virtually impossible to guarantee that the state information is accurate. SST overcomes these difficulties using a database paradigm.

The SST application uses a database to store the information about users and the trips they are registering. (We use the mSQL database written by David Hughes available as shareware at: <ftp://bond.edu.au:/pub/Minerva/mssql>). This cumulative information (trips and contact details) makes up the state information about a user. Figure 2 is a complete state diagram for the SST program. The states are represented by ovals and the state transitions by arrows. Users always transit between states in the direction of the arrows.

Each of the transitions is marked with the abbreviation for one or more of the 14 specific routines that must be invoked to accomplish individual state transition. If more than one routine is needed, several program abbreviations appear. SST was designed modularly so that a fixed set of routines is applied in appropriate order to implement the overall state transition process.

We use database relations to maintain state information during transactions. A transaction in the SST program is the action of moving from one state to another. To preserve the state information, the SST application uses two sets of database tables: (1) the permanent tables that exist as a file and (2) temporary tables created when a user registers or logs in. The user information is updated in the temporary tables from the WWW pages, and when specific state transitions take place (specifically, those labeled DIQSG in Figure 2), the temporary tables are written to the permanent tables. In this way the user information is saved across state transitions. SST is implemented as HTML with embedded forms. The usual method for handling a form embedded in an HTML document is to create a custom, common gateway interface (CGI) program that performs any actions required by the form.



**Legend**

- Q Query
- D Delete
- S Sort
- G Group
- Qm Match Query
- Io Initial/uses info
- I Insert

**Data Management**

	IN	OUT
PT Print Trips	(List)	(Trips in List)
PGT Print Grouped Trips	(List)	(Trips in List)
EM Email		
PG Print Group	(Number)	
PPS Package Per Screen		
DTT Delete Temp Tables		
DU Delete User		
MQR Match Query Report		

**Data Base**

- QSG (Filename)
- DIQSG (Filename)
- Qm (List, Parameter)
- Io (Name, Password....)
- DFM Data File Manager (List, Variables)
  - {Purge Date, Inception Date, Round Trip (flag), Return Day (flag), Day, Type, t<sub>d1</sub>, t<sub>d2</sub>, t<sub>a1</sub>, t<sub>a2</sub>, P<sub>d</sub>, P<sub>a</sub>, Group#, Trip#}
- DF Data File

Figure 2: SST state diagram

State transitions are then implemented as the invocation of the CGI Form Server by an HTML form.

Because the state transition takes place only on the submission of a form to the CGI Form Server, several WWW pages may exist within one state. For example, when selecting trip origins, the user is in the state labeled Origin in Figure 2, and the entire location tree is traversed while in the Origin state. The CGI Form Server is invoked only after a site at the base of the tree is chosen, and the trip origin coordinates are then committed to the database. The act of committing the data to the database causes a transition to either the Destination state (where trip destination coordinates are collected) or to the Fine Tune state.

## RESULTS

The SST program went on-line mid-March, 1996. In this paper we present the results as of mid-November, 1996. The number of users participating in a ridesharing program can have a significant influence on the success of such a program; however, the absolute number of users needed to be successful is difficult, if not impossible, to quantify. Figure 3 presents the number of UW users acquired by the SST program as a function of time; it also presents the same statistic for the long-term, regional carpool effort operated by King County METRO<sup>1</sup>.

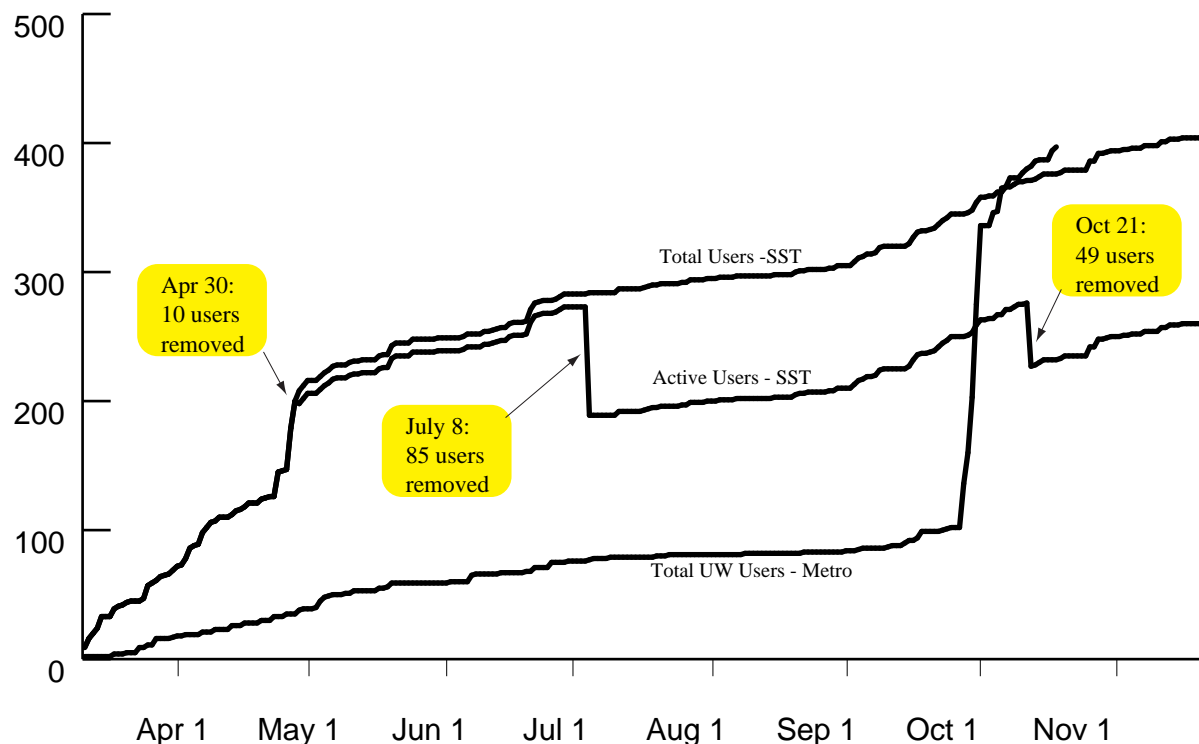


Figure 3: Cumulative and active users as a function of time

<sup>1</sup>Note that we are comparing the newly acquired users not the total user database. KC METRO began the study period with some number of existing users.

While both ridesharing efforts end the study period with roughly equal numbers of new users, several important differences exist. The first difference is the clientele; in comparing the users (by name), there is only a 20% overlap between the two efforts, indicating that SST is reaching a clientele that did not exist for a long-term, regional carpooling effort. The second difference has to do with the service provided to new users over the study period. If the area under the user curves is viewed as user-weeks of service, the SST effort provided substantially more service over the period investigated. This additional service results from the high availability of SST to the user. Direct user access, 24-hours a day, is a feature of SST not found in more traditional ridematch systems. Figure 4 presents SST use by time of day. To date, almost 20% of the system accesses have occurred before 8:00 am or after 5:00 pm, indicating a sizable portion of the system usage occurs outside of normal business hours. This again indicates that there is a niche for a direct-access ridematching system which is unmet by traditional ridematch efforts. Based on these two observations, SST reaches a new, sizable group of users compared to an established carpool system.

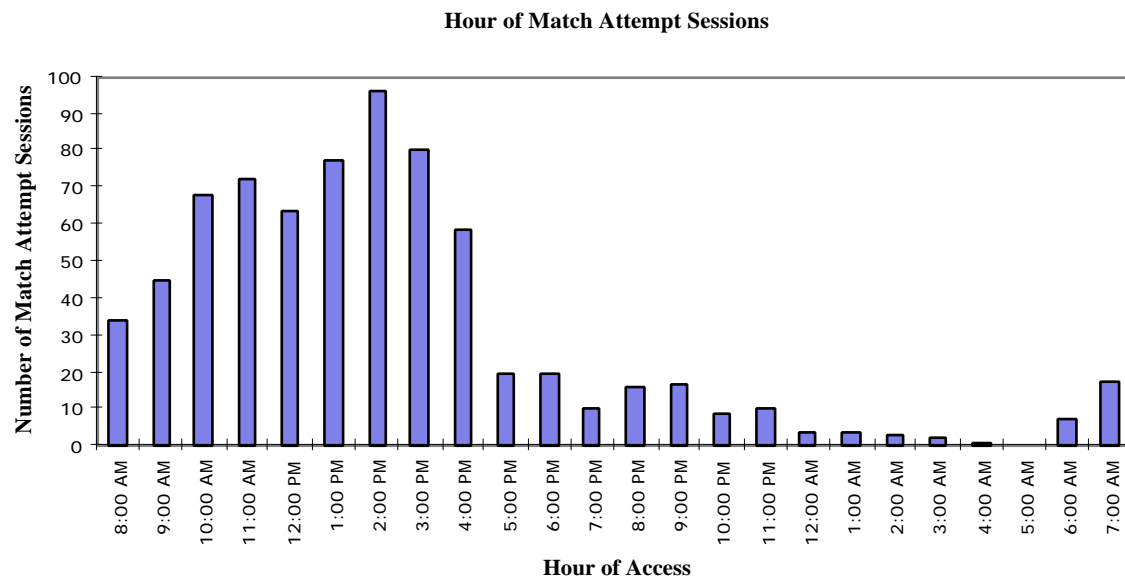


Figure 4: SST hours of use

SST was designed to operate in an environment with variable schedules and was implemented at the UW which operates on a ten-week term system. In the face of schedules that can be drastically different between terms, effective ridematching must guarantee that the trips being matched really are the current trips that the users might share. To address this problem, SST has the notion of active participants. An active participant is a user that has accessed SST during the current

academic quarter, typically to update schedule information or to seek a ride. Active participants are the only users that are eligible to be matched with other users (the users who have not accessed the system in the current quarter are inactive and not used for matching purposes). Figure 3 shows the active users as a function of time, as well as the number of users purged at term boundaries. While purging of inactive records reduces the total number of users in the matching database, it increases the utility of the information. It is noteworthy that a significant number of the SST users remain in the active category even after three purges, indicating that the ridesharing community reached by SST is an active one that is fairly stable over the study period.

For ridesharing to be successful, the participants must have trips with matching characteristics. Figure 5 presents the temporal history of: (1) the cumulative number of attempted matches, (2) the cumulative number of successful matches, and (3) the cumulative number of email messages sent to attempt to establish a carpool. Trips entered by SST participants quickly reached a match rate of approximately 39% and maintained that match rate over the study period, even with a relatively small total population and variable schedules.

Some other interesting statistics come out of the SST data. As of the writing of this paper, 68% (112 people) of the active users were members of the faculty and staff of the UW; 32% (53 people) were students. Among the staff user group, 90% (101 people) have regular and unvarying schedules typical of traditional ridesharing while only 10% (11 people) indicated any variation of schedule over the course of a week. Among student users, 66% (35 people) have traditional schedules, while 34% (18 people) have schedules which vary by day. Of the 2,065 trips registered in the data base, 93% of these trips are categorized by the users as traditional commute trips, 4% of all the registered trips are user-categorized as recurring, non-commute types, and only 3% are identified as special dynamic trips, the modality at which this study is targeted. The number of student users and variable schedules has grown considerably since an outreach campaign was begun in the fall of 1996 to attempt to involve larger segments of the student body. The correlation between increased student use and increased numbers of dynamic trips leads the authors to believe that as the student user group continues to grow, the dynamic capabilities offered by the SST program will be increasingly utilized.

The discussion presented in this paper reflects the quantitative measures that can be observed internal to a ridematch system. The larger question of formation of carpools requires direct interaction with the users in the form of survey instruments. In the coming months, surveys will be distributed to the users to obtain quantitative information on carpool formation based on ridematch results. However, to date, SST has demonstrated that there is a sizable ridematching clientele, who are outside the existing traditional carpool community, and whose needs are met by a dynamic, direct-access, widely available, instant ridematching system like SST.

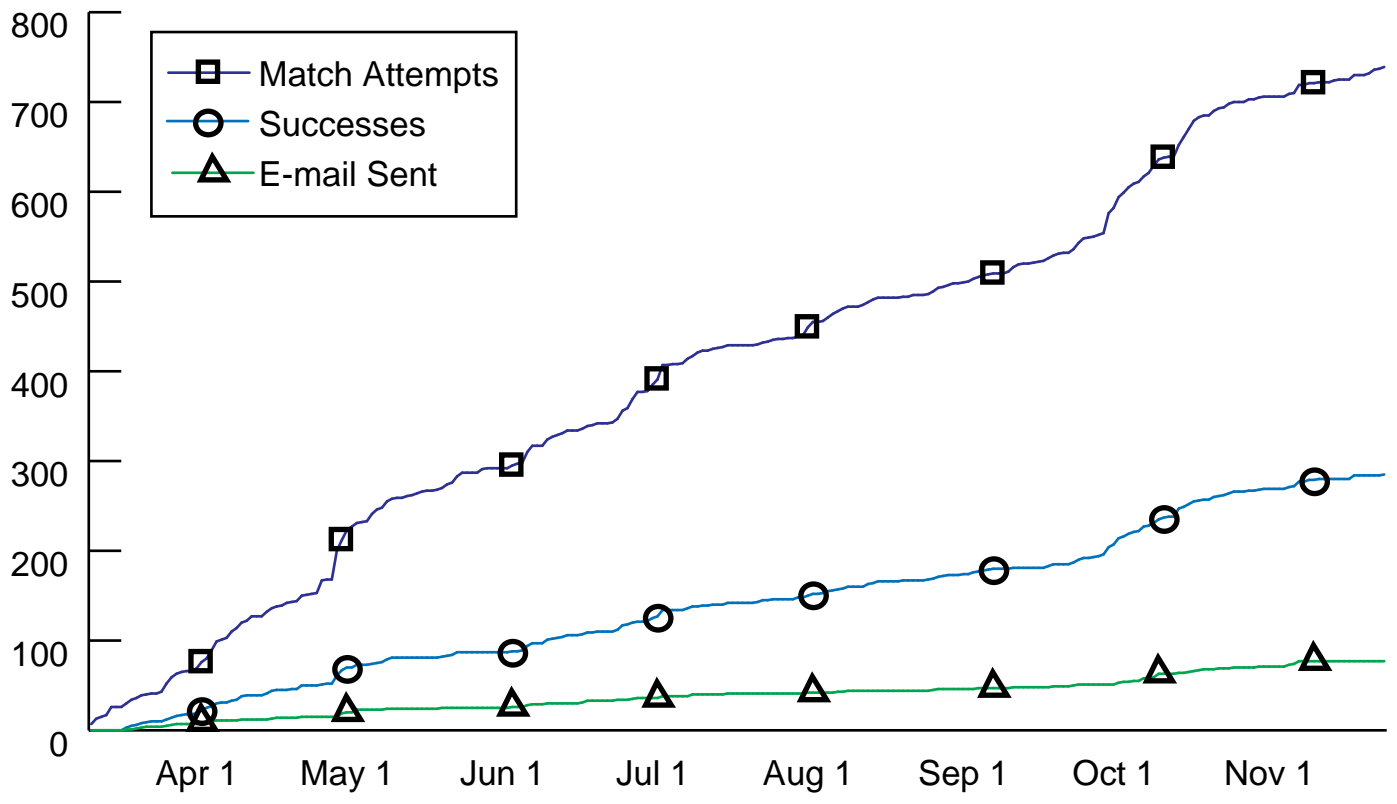


Figure 5: Usage as a function of time

See SST at <http://sst.its.washington.edu/sst>

## REFERENCES

- [1] Ayland, P.D., D. Hill, S. Rutherford, M. Hallenbeck, and C. Ulberg. *Assessment of Advanced Technologies for Relieving Urban Traffic Congestion*. National Cooperative Highway Research Program Report 340. American Association of State Highway and Transportation Officials, FHWA, December 1991.
- [2] Casey, R.F., L.N. Labell, R. Holmstrom, J.A. LoVecchio, C.L. Schweiger, and T. Sheehan. *Transportation Demand, Management Technologies*. Chapter 5 of *Advanced Public Transportation Systems: The State of the Art, Update '96*. Report No. FTA-MA-26-7007-96-1. U.S. DOT, FTA, Office of Mobility Innovation, January 1996, pp. 109-139. URL: <http://www.fta.dot.gov/library/technology/APTS/update/CHAP5.HTM#>
- [3] Haselkorn, M., J. Spyridakis, C. Blumenthal, S. Michalak, B. Goble, and M. Garner. *Bellevue Smart Traveler: Design, Demonstration, and Assessment*. Report WA-RD 376.1. Washington State DOT, FHWA, 1995.
- [4] Koppelman, F.S., C.R. Bhat, and J.L. Schofer. *Market Research Evaluation of Actions to Reduce Suburban Traffic Congestion: Commuter Travel Behavior and Response to Demand Reduction Actions*. *Transportation Research, Part A (General)*, Vol. 27A, No. 5, 1993, pp. 383-393.
- [5] Michalak, S., J. Spyridakis, M. Haselkorn, B. Goble, and C. Blumenthal. *Assessing Users' Needs for Dynamic Ridesharing*. In *Transportation Research Record 1459*, TRB, National Research Council, Washington, D.C., 1994, pp. 32-38.
- [6] Misch, M.R., J.B. Margolin, D.A. Curry, L.J. Glazer, and G. Shearin. *Guidelines for Using Vanpools and Carpools as a TMS Technique*. National Cooperative Highway Research Program Report 241. American Association of State Highway and Transportation Officials, FHWA, December 1981.
- [7] Puget Sound Council of Governments. *Transportation Operators' Committee. Regional Ridesharing Assessment and Recommendations*. The Council, Seattle, WA (216 1<sup>st</sup> Avenue So., Seattle, WA 98104), 1988.
- [8] Reddy, P. DVG. *GIS Based Real-Time Rideshare Matching*. URL: <http://wwwsgi.ursus.maine.edu/gisweb/spatdb/gis-lis/gi94082.html>, 1994.
- [9] Stevens, W.F. *Improving the Effectiveness of Ridesharing Programs*. *Transportation Quarterly*, Vol. 44, No. 4, October 1990, pp. 563-578.
- [10] Walbridge, E.W. *Real Time Ridesharing Using Wireless Pocket Phones to Access the Ride Matching Computer*. *Pacific Rim TransTech Conference. 1995 Vehicle Navigation and Information Systems Conference Proceedings*. 6<sup>th</sup>

*International VNIS. July 30 - August 2, 1995, Seattle, Washington, United States, pp. 486-492.*