A Train Position Monitoring System Based on COTS and Free Software Components

D. W. Carr, R. Ruelas
Departamento de Ingeniería de Proyectos, Universidad de Guadalajara
Apdo. Postal 307, C.P. 45101 Zapopan, Jalisco, Mexico
doncarr@gmail.com, rruelas@newton.dip.udg.mx

Raul Aquino Santos
Facultad de Telematica, Universidad de Colima
Colima, Colima, Mexico.
aquinor@ucol.mx

Apolinar Gonzalez Potes
Facultad de Ingeniería de Mecanica y Electrica, Universidad de Colima
Colima, Colima, Mexico.
apogon@ucol.mx

Abstract

This work presents a position monitoring system to generate displays showing the position of trains, time between trains, seconds early/late for each train, and alarms. Alarms will be generated when trains are too close, unexpected events are received, or trains do not cross subsequent sensors when expected. This system has been designed to run concurrently with the existing original system, which only illuminates LEDs on a large plastic board to indicate the segments that the trains are in. Currently, it is up to the operators to estimate times between trains by viewing the panel where lights are illuminated. The new system will greatly improve performance issues in addition to the safety issues. In accordance with the agreement between the Universidad de Guadalajara and the light rail system (SITEUR), the license for this project will be GPL and it will be available on *sourceforge.net*

1 Introduction

We are in the process of creating a new train position monitoring system, based on COTS hardware and free software components for the light rail system in Guadalajara, Mexico. The light rail system in Guadalajara has two parallel tracks, where the trains travel from one extreme to the other, and then return on the opposite track as shown in Figure 1. We have also developed a simulator, separate from the monitoring system, that creates crossing events just like a real running system would. With this, a real running system can be switched for the simulator, with no changes in the position monitoring system.

The position monitoring system generates displays showing the position of trains, time between trains, seconds early/late for each train, and alarms. There are two types of screens generated. The first, as shown in Figure 2, is a graphical display where the position of each train is shown to the nearest 1/4 of the distance between stations. The second, as shown in Figure 3, is a tabular display, with columns for train id, time in service, seconds late, percent late, time to the next train, last departure station, time since departing, and an approximation of the number of passengers currently on the train.

This system will run in parallel with the existing original system, which only illuminates LEDs on a large plastic board to indicate the segment that each train is in. Currently, it is up to the operators to estimate times between trains by viewing the panel

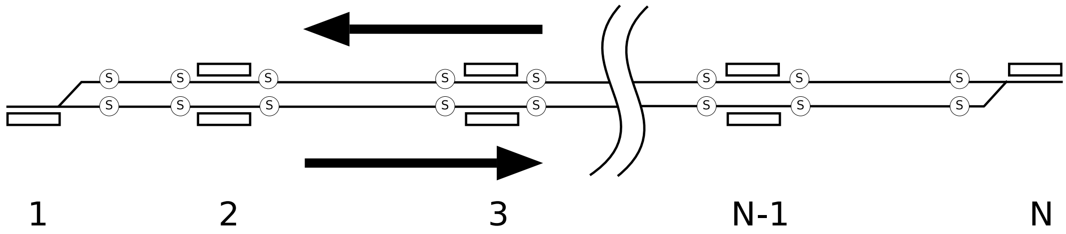

FIGURE 1: *An Example Train Line*

where lights are illuminated. Given the number of trains and the inexactness of the information, it becomes a nearly impossible task, and thus it is currently very likely that operators in central control would miss dangerous situations. The new system will automatically calculate the time between trains based on the time of the last sensor crossed. Between sensors, the position of the train will be estimated, and then corrected every time another sensor is crossed.

Actually, monitoring the spacing between trains is a performance issue as well as a safety issue, and spacing problems become performance problems before they become safety problems. So, there will typically be performance based alarms long before any safety related alarms are triggered. Alarms will be generated when trains are too close, unexpected events are received, or trains do not cross subsequent sensors when expected. Maintaining the spacing between trains for quality of service issues is also exactly what is needed for safety.

This system will start out as a position monitoring system, but is designed to be upgraded to a full signaling system eventually and replace the existing signaling system.

We are using various programming techniques including N-Version programming[1][2][3], Internal instrumentation to detect errors, design patterns, peer review, etc, to provide the level of reliability required for safety critical systems.

2 Hardware

All hardware used for this project will be off-the-shelf. Where crossing sensors are located there will be either a commercially available RTU or single board computer running GNU/Linux connected to a TCP/IP network. These remote computers will send crossing events to central control. In central control, there will be high reliability servers running GNU/Linux as well. To start, the network connecting the remote devices to central control will be a single standard TCP/IP based network that will almost certainly not have the MTBF required for a safety critical system, but will be satisfactory for testing and development. It is expected that the current network will be replaced by two redundant loop networks to achieve the MTBF required for a safety critical system. Each loop network can be cut at any single location, and continue working, with the traffic re-routed automatically. The system as a whole can continue to function as long as one of the loop networks is working. As long as there is a very active maintenance policy in place to immediately fix any detected faults this system will deliver an MTBF at the levels required. It should be noted that all servers and remote devices will have two network connections so that they can all be connected to two networks.

3 Software

The software for this project is based on high volume free software components such as the Linux kernel, as well as the specific application written in C/C++ and also available as free software under the General Public License. The software currently consists of two modules that communicate via message queues. The first module is the position monitoring system that currently generates the displays for the operators in central control, and the second module is the simulator that generates crossing events just like a real running system. When the final hardware selections are made for the devices located next to the crossing sensors, we will write a driver to communicate with them and relay crossing events to the monitoring software using the same communications as the simulator, so that we can switch the simulator for a real running system.

3.1 Position monitoring module

The position monitoring module will generate a graphical display as shown in Figure 2, and a tab-

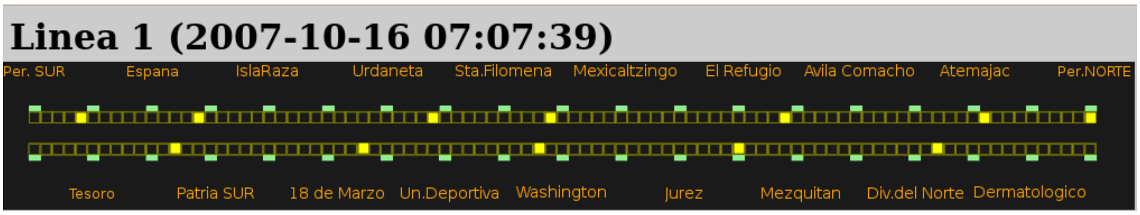

FIGURE 2: *Graphical Display*

ular display as shown in Figure 3. Based on feedback from managers at the light rail system, we will also generate alarm windows for both performance based alarms and safety based alarms. For now, this is only a monitoring system, and it will be up to the operators in central control to act on information provided, and use the existing radio system to call the drivers and ask them to stop the train, or take whatever actions are necessary to maintain the safety of the system.

3.2 Simulator module

The simulator module generates events just like a real running system. At start-up, it reads the actual time table for one of the lines at the light rail system in Guadalajara, and uses this information to simulate a complete day of crossing events. For line 1, there are now 156 departures per day, with each train making a round trip, returning back to the originating station at Periferico Sur. There are 19 stations, generating arrival and departure events both for the trip up, and back, generating roughly 12,000 crossing events per day. We typically run the simulator at 10x real-time for demonstration purposes since real-time is painfully slow. The simulator however, can be run at any speed (within limits of processing time on the target hardware) to simulate a whole day in a matter of seconds for testing purposes. We can use this then to simulate a variety of dangerous situations and verify that the monitoring module responds correctly and generates the appropriate alarms at the correct time.

3.3 Driver module

The driver module will be a drop-in replacement for the simulator, and feed the position monitoring module with the actual crossing events from a real running system. The driver module will communicate via a fibre optic network with actual devices connected to the crossing sensors, and relay crossing events to the position monitoring module. We plan as a minimum to have arrival and departure sensors for both directions at each station, thus monitoring on the order of 76 sensors connected to 19 remote devices for line one of the light rail system. On average, there will be one crossing event generated every 6 seconds that is sent to the position monitoring module for processing.

3.4 Software reliability

We have discussed software reliability issues in other articles [4][5][6].

We are using n-version programming, internal consistency checks, design patterns, world-wide peer review, etc, to get the level of reliability required for safety critical systems. The first system is only a monitoring system, to be run in parallel with existing safety systems, but we realize that operators in central control will come to trust and depend on the information generated to keep the system safe. Thus, this must be considered a safety critical system, and we are treating it as such, and, in the future, this could possible be expanded to be a full signaling system. At the same time, we are also working on systems to add additional layers of systems to work in parallel with this system to provide even more safety. We are currently experimenting with transceivers that can be used for collision avoidance, where trains will be able to connect with each other when they are close enough, and begin exchanging speed and location information, thus enabling algorithms that would stop a train that is dangerously close to the train in front. We are also working on on-board systems that would have a subset of full CBTC features so that would monitor speed and position, and allow commands from the central system to stop trains when dangerous situations are detected. Of course we are hoping for world-wide review to help generate safety critical systems for the remaining parts of the world that do not have the money for the systems developed for the industrialized world.

Linea 1 (2007-10-16 07:18:56)							
Tren ID	**Tiempo en servicio**	**Tarde segundos**	**Porciento**	**Tiempo a Siguente**	**Ultimo Salida**	**Timpo desde Salida**	**Numero de Pasajeros**
20	00:11	0	-	05:07	PerifercoSur	11	57
19	05:26	(8)	(2.5%)	05:02	PatriaSur	(en estación)	165
18	10:41	4	0.6%	05:33	Urdaneta	(en estación)	278
17	15:54	(17)	(1.9%)	04:40	Washington	28	345
16	21:08	16	1.3%	04:54	Refuigo	25	281
15	26:26	39	2.5%	05:53	DivNorte	5	163
14	31:41	0	0.0%	06:09	(PeifercorNorte)	(en estación)	69
13	36:56	(55)	(2.5%)	04:58	(DivNorte)	18	185
12	42:11	(39)	(1.5%)	03:25	(Refugio)	(en estación)	282
11	47:26	70	2.5%	06:24	(Mexicaltzingo)	(en estación)	338
10	52:41	0	0.0%	06:42	(Urdaneta)	(en estación)	251
09	57:55	(89)	(2.6%)	03:25	(Espana)	4	144
08	63:11	21	0.6%	-	(Tesoro)	94	78

FIGURE 3: *Tabular Display*

4 Conclusions

It is the opinion of the authors that a train position monitoring system can be created based on COTS hardware and free software components that is more appropriate for the conditions in developing countries than current proprietary systems costing tens or even hundreds of millions of dollars. We have in fact created the the first modules for the light rail system in Guadalajara Mexico, and have a full monitoring system and simulator running as proof of concept that will allow us to work with the managers at the light rail system to determine the specifications and features for the final system. This will improve the safety over the existing system that only illuminates lights on a board to show the location of each train. It is currently nearly impossible for the persons in central control to keep track of all trains in the system and know when dangerous conditions exist. The new system will be able to generate the position of all trains in both graphical and tabular format for operators to easily visualize the location of the trains and respond to dangerous situations. We will also automatically generate performance based and safety based alarms.

References

[1] A. Avizienis, 1995, *The Methodology of N-Version Programming*, Software Fault Tolerance, Ch. 2, John Wiley & Son Inc, pp23–46

[2] J. Dugan and R. Lyu, 1994, *System Reliability Analysis of an N-version Programming Application*, IEEE Transactions on Reliability, Vol. 43, No. 4, pp513–519

[3] L. Hutton, 1997, *N-version design versus one good version*, IEEE Software, November/December 1997, pp71–76

[4] D. W. Carr, R. Ruelas, J. F. Gutierriez-Ramirez and H. Salcedo-Becerra, 2005, *A Communications Based Train Control System Using a Modified Method of N-Version Programming*, Congreso de Instrumentación (SOMI XX), León, Gto., Mexico, 2005.

[5] D. W. Carr, R. Ruelas, J. F. Gutierriez-Ramirez and H. Salcedo-Becerra, 2005, *An Open On-Board CBTC Controller based on N-Version Programming*, Proc. IEEE International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA'05), Vienna, Austria.

[6] H. Salcedo-Becerra, D. W. Carr, R. Ruelas, G. A. Ponce-Castaeda, 2006, *Performance Monitoring for the Light Rail System in Guadalajara, Mexico* Int. Conf. on Dynamics, Instrumentation and Control, August 13-16, 2006, Querétaro, Mexico.