CFS Documentation - This Version started 13 March 2019

A More Prototypical Approach to Car Forwarding Software

This is the comprehensive documentation for the program, from which will be extracted the parts for the clinic.

Abstract, Version 1.1

"A More Prototypical Approach to Car Forwarding Software"

Traditional computer based approaches to car forwarding on a model railroad can easily generate lots of activity, however the car movements are often random rather than being along a specific route to fulfill a prototypical shipment. These unrealistic car movements are a common criticism of such programs.

Individual Car Cards and Way Bills (CC/WB) on the other hand, can easily represent the richness of prototype shipments, but they require a lot of manual work to create, modify, use, and set up between operating sessions.

This clinic will discuss the progress on a new computer program I have written to overcome both of these limitations. The goal was to maintain the richness of the prototypical shipment based CC/WB system while eliminating all of its manual set up work between sessions.

By simulating multiple days, it is easy to compare different operating scenarios as it is all just data inside a computer program. This is very useful for tuning car traffic on an existing layout, and can even influence new or expanded layout design by simulating a layout before it is even built.

Last Updated: 29 November 2018

Background

Numerous software programs have been developed over the years to generate waybills and/or switch lists for controlling how cars move over a layout during an operating session. Unfortunately the logic behind many of these is not based on how real cars come to be moved in trains. There is nothing random about prototypical car movements, every one is done in response to some specific desire to move a shipment of some commodity from one place to another. However, given the complexity and scope of the real world and the variety of shipments, shippers, and destinations, a snapshot view of the car movements over any small portion of a railroad will often appear to be random, even though in reality it is not. The only thing that could be considered random is the particular car selection by the railroad to satisfy the shipper's request. Almost all other aspects, such as car type, length, owner, class, etc. is specified by the shipper, or by some other convention.

Any computer program that selects car movements using some random process cannot simulate prototypical car movements. While this approach will often generate sufficient amounts of traffic and movements that can mimic the appearance of the prototype, such approaches will not stand up to a closer scrutiny of where the cars are actually going and why.

This report describes the work I am doing to create a completely new program that simulates where cars move based entirely on specific commodity shipments that have been specified between given shippers and receivers. The only randomness in the program is the selection of a particular shipment for a car from a list of otherwise equal candidate shipments for that car. In other words, a given boxcar may be able to satisfy many different shipments to a variety of destinations on the layout. In the absence of any other deciding factors, the program will select one of those candidate shipments

at random. All other aspects of how the program works are strictly not random but follow prototype practice as closely as possible. Once a car has been assigned to a specific shipment, waybills are generated and trains will then move that car according to those waybills until the car is released or it returns to staging where it effectively completes its journey and is released.

All aspects of which train will pick up that car, the direction the car travels, how long it remains to be loaded, etc., are deterministic and not random. Routing, for example is determined by the arrangement of the network of tracks between the shipper and the receiver. Normally the shortest distance will be selected from multiple routes but there are techniques to control this.

In many ways the program approaches car movement in the same way as Car Cards & Waybills do, but without the manual effort. The output from the program is a set of Switch Lists for the crews to follow. Yard switching uses a special format, but the underlying principles remain the same.

Motivation

The primary motivation behind developing this software was the elimination of the tedious manual work of setting up paper Car Cars and Waybills (CC&WB) while preserving its rich and prototypical shipment based car movements. I do not enjoy all of the paperwork required by traditional CC&WB systems, so I want a computer to handle that while still producing realistic car movements and easy to follow switch lists.

The best way to think of what it does is that the computer program takes care of all of the clerical work to select and assign Waybills to Cars, and to then generate Switch Lists for each train from those Waybills.

The focus should be on running trains and switching cars and not on the paperwork required to do that. While I appreciate that the paperwork is all very prototypical, it does not appeal to me so I want to automate it.

A second major goal was to be able to use the program to simulate the layout operations over multiple days and produce a statistical analysis as a layout design tool.

Primary Objective

Simply put: To have the prototypical richness of CC&WB with none of the manual effort needed to set up a layout prior to an operating session.

Objectives in Detail

The program will:

Decide where all cars will be moved to and what trains will move them.

Base all car movements on prototypical shipments and waybills, with no model railroad specific assumptions, approximations, or random peculiarities.

Route cars based on the layout topology rather than information entered manually.

Handle complex car movement sequences involving multiple trains as with CC&WB.

Re-stage trains in staging so that there is no manual intervention with the cars required.

Eliminate the need for any manual set up of the layout before an operating session.

Allow different train schedules to be used to accommodate different operator experience levels.

Produce switch lists that are easy to follow for guests and infrequent operators, while still providing enough challenge for experienced operators.

Include all of the relevant information on the switch lists that is required without the need for the operator to consult additional documentation.

Analyse the operations over multiple simulated days producing various statistics to provide insight into traffic patterns, congestion, etc., that can be used for planning a layout in advance.

Simulate time as being continuous with no arbitrary operating session divisions.

Run on all major computer platforms.

High Level Overview

So, how does it work?

There are two main parts to the program. One selects shipments for cars and generates waybills, while the second runs trains that pick up and deliver the cars according to those waybills.

A good example is a train in staging that is about to return to the layout. First, each car must be assigned a suitable waybill so that it can be moved onto the layout, either for delivery to an industry as a load or empty, or to just pass over the layout into a different staging area. Second, the train is run over its route and cars are picked up and delivered according to their waybills.

While this all sounds simple, the trick is in selecting appropriate shipments for the cars, based on a number of considerations, so that the train can in fact advance those cars.

Data Entities

The program data mimics real world entities. These include:

Railroad – The owner of the Cars and Stations.

Station – A named location on a Railroad that can contain any number of Industries.

Industry – Part of a Station that can contain any number of Tracks.

Track – Part of an Industry that can accept Cars and has any number of Shipments to or from it. Trains visit Tracks to pick up and deliver Cars. The fully qualified name of a Track includes its parent Railroad, Station, and Industry.

Car – A piece of rolling stock that has a particular Car Type, length, owner Railroad, etc. and can be loaded and moved from one Track to another Track by a Train.

Car Type – Denotes the kind of Car, such as Boxcar, Tank, etc.

Load – Loads are carried by a Car and required by a Shipment. Can be defined as anything, such as Lumber, Oil, etc. The "Empty" load is always available.

Waybill – Controls where Cars will be moved to. There are three types of Waybills, Empty Car Bills, Freight Waybills, and Return to Home Bills. They all serve the same purpose of controlling the routing of a Car between where it is picked up by a Train to where it is delivered. Each Waybill gets implemented through a series of CarMoves that are honoured by the Trains.

Car Move – Details where the Car should be moved to by a Train to complete the parent Waybill.

Timetable - A sequential list of Timetable Stops. Trains will run according to a Timetable.

Timetable Stop – One Track along a Timetable with a relative time when a Train is due there along with an operation to be preformed by the Train, such as a pick up or delivery.

Train – An engine and caboose that will follow a Timetable, visiting Tracks and picking up or setting out Cars. A Train has a specific start time for its Timetable.

Switch List – A printed list of all of the operations to be done by one Train.

Shipment – A particular Load, using a particular Car Type that is being shipped from a shipper Track to a receiver Track.

Key Concepts

Date and Time

- All time is continuous and modelled using real time. There is no concept of "fast time" within the simulation, but the use of a fast clock during sessions is often a good idea.

The notion of a "session" requiring set up beforehand does not exist. Each operating session with people simply starts where the previous session left off and runs as long as everyone wants.

Car Wait Time

- Cars have a wait time applied to them by the Waybill when they are delivered to an industry. This represents the time that the car must stay there while it is loaded or unloaded, before it is able to be moved again. This wait time is not fixed and is controlled by rules that are followed when the car's waybill is created. The time can be short to simulate icing a reefer, or can be as long as needed to load a complex shipment.

Modelled and Unmodelled Locations

- All shipment locations are included in the data, but some are not modelled as they represent far away places behind staging. At least some representation of these is required in order to support any shipments going to and from them. If desired, a great deal of realism can be added by defining significant unmodelled portions of the real railroads behind staging. The only real benefit to doing this is if one wants to see the complete routing to or from a remote industry. None of this really matters are cars only come from and return to the staging tracks.

Car Capacity

- Track and train maximum capacities are observed so that there are never more cars than will fit in a given location. This is considered when trains try to pick up cars, and when shipments are being selected for restaged cars. It may be desirable to add the ability to "overbook" some locations, in order to give the crews a little extra to deal with. More experimentation will be needed for this.

- Shippers may send more cars to a destination than it can handle and cars have to be spotted temporarily elsewhere. This situation probably arises in simple model car forwarding systems due to the nature of the random, uncontrolled assignment of cars to loads. In the real world, shippers, receivers, and the railroads are much more in touch with what is happening than a simplified random process used on a model railroad. In order to avoid this situation, the default behaviour of the software is to not schedule car movements that would overload either a destination or a train's capacity. It may be desirable to allow this restriction to be relaxed somewhat so that there will be occasional overcrowding that must be handled by the crews. This can be added later.

Shipment

- All cars move to satisfy specific shipments. Shipments can be of one of four types:

Inbound - Unmodelled shipper to modelled receiver

Outbound - Modelled shipper to unmodelled receiver

Through - Unmodelled shipper to unmodelled receiver

Local - Modelled shipper to modelled receiver

Waybill

- There are three main types of waybills used:

Empty Supply Bill

Freight Waybill

Return Bill ?

Empty Supply to a shipper, Freight from a shipper to a receiver, and Car Return from a receiver.

All three types can be generated for the full life cycle of a car, but some get ignored for cars coming from or going to staging. For example, a car in staging that gets restaged as an inbound load will not have the Empty Supply waybill generated for the car to be first delivered to the unmodelled shipper.

Additional types of waybills can be added if needed. It is also possible to create a specific waybill by hand, but this capability has not been added.

Car Order

- Specific industries can make requests for cars and these will take precedence when deciding what shipments should be applied to available cars. Prototype car orders are always for an empty car to be loaded, however in the program car orders can also be for an inbound load. In this case, the loaded car order can be considered as the culmination of a regular empty car order placed by some off-layout shipper, resulting in a load being delivered on the layout. Car orders will affect the overall diversity of car movements and should only be used for specific scenarios to achieve a desired car flow pattern at certain industries. Places like team tracks should probably be best left to more random shipment selection to enhance diversity.

Car Selector

- Both shipments and car orders include a car specification to influence which specific car gets selected. This cannot be only a rigid precise specification, as that is not prototypical. For example one shipper may not care how wide a boxcar door is, but it may be critical for another. Therefore the specification is composed of many parts, such as car type, door width, length, owner, etc., however each of these can be represented by a “wildcard” to indicate that anything will be considered a match.

Randomness

- No aspect of the program's operation is random, except for one. The selection of a candidate shipment for a given car is randomly chosen from the set of only those shipments that are suitable for that car and the current state of the layout. In other words, if there are three suitable shipments for a car which are all otherwise equal, then one will be selected at random.

Staging Tracks

- These are special and form the boundary between the modelled and unmodelled portions of a

layout. All cars leaving the layout to a staging track are assumed to continue on to their final destinations, and all cars returning to the layout from staging are assumed to be continuing on from their original starting point.

Restaging

- Every train that is in staging must have new waybills assigned to all of its cars before it can return to the layout. Each car is therefore restaged by assigning it a suitable waybill. Any cars that cannot be assigned a waybill are removed from the train and placed in storage until such time as there is a suitable waybill available.

Layout Topology

- All tracks in the layout are connected together to form a network of both modelled and unmodelled tracks. Paths from one track to another are all calculated by the program.

Car Routing

- The layout network paths are used to determine the routing of a car from pickup to delivery. A car must follow this route when being picked up and delivered by trains.

Car Reblocking

- Reblocking consists of rearranging the cars within a train into blocks that share a common destination. This makes switching in a yard and along the line at a town easier as it reduces the number of moves required to get the cars where they need to be. The program currently has a crude reblocking strategy that proves the concept, and as with many other features, the program’s modular design will allow for this to be substituted with better strategies in the future. For now, it works and has proven the value of having the feature.

Layout State

- The state of a layout is the topology of the tracks plus the current locations of all of the cars with their pending waybills. This is for a specific point in time. There can be no active trains when a layout's state is saved.

Study

- A study consists of a sequence of activities such as running a train that can be applied to the current state of the layout to produce a new state and associated output such as switch lists. Each of these states is stored separately so that it is easy to rerun a study with different parameters, or to run different studies to compare the results.

Train

- Train pickup and delivery of cars parallels how it is done with traditional Car Cards & Waybills, in that each train considers the cars it encounters along its route and decides if it will pick them up or not, based on the car's waybills which indicate where the car wants to go next.

- Trains follow a specific timetable which specifies where they go and what they do. A train will only pickup a car if it can advance it towards the car's final destination. Usually more than one train is required to advance a car over its complete route. Multiple trains can operate with overlapping timetables and all operations are done in time sequence. Different trains can use the same timetable, but with different starting times.

Car Advancement

- Cars effectively "hitchhike" their way along, according to their waybills. Any train that tries to pick

up a car will do so only if the train can also deliver the car somewhere further along the car's route, meaning that a number of criteria must be met, such as there being room for the car, the train's routing, etc.

Misplaced Cars

- If a car does not get delivered to the correct location on the layout, it is easy to generate an extra waybill to add to the car to move it from where it is to the correct location.

Timetable

- Specifies the relative times that a train will visit a location and what action it will do at that location. One is required for all trains, not just the "official" timetable trains. The timetables are also used to produce a string line diagram.

Event

- During the running of a study, all changes are recorded as events, such as car pickup, delivery, etc. These are then processed to produce switch lists and various other forms of statistical analysis.

Switch Lists

- Two formats are used for switch lists. The first is for road trains and is a traditional list of locations and the work to be done at them. This does not work very well for yard switchers that are working in one concentrated location, as the order of the car moves cannot be predicted and crews are free to sort that out for themselves. A second format used in yards simply lists where all of the cars are currently at, and where they need to be moved to by a certain time for other trains to pick them up. The yard crews will figure out the best way to accomplish this. These also include lists of the cars being delivered and picked up by passing road trains so that the yard crew can be ready.

- Switch lists are generated for every train so it is easy to include special instructions that pertain to that train on that day, instead of being just generic instructions in an additional book that the crews must also understand.

Analysis

This needs to be edited a LOT more.....

- The program can be used as a planning and analysis tool for both existing and future layouts. It can generate copious quantities of historical and statistical data from a study, and these can be reported in tabular and graphical formats. For example, a fundamental insight is if there is anywhere on the layout that cars tend to accumulate because there are not sufficient trains to advance them to where they need to go. A simple plot over time of the number of cars on a track or at a station will clearly show if the cars are flowing as expected or are accumulating unrealistically. By relaxing various capacity limits significant insight can be gained into the dynamics of the layout without having to wait until many operating sessions have been completed, and then only rely on the crew’s impressions.

- Another useful insight is a list of all of the places that a given car has visited over many days. This can help show if there is not sufficient diversity of shipments for that car type, in that it only goes to certain locations.

- Other statistics can show how often specific shipments get selected, minimum, average, and maximum daily track occupancy, and so on. More can be added as ideas and needs emerge.

- Currently, all reports are produced from a variety of output text files that are read and manipulated by a spreadsheet program to do calculations and plot charts. These will eventually get replaced with

integrated reporting capabilities inside the program.

- A number of graphs and other statistics are produced to help understand how well the studies are simulating a balanced railroad operation. (Need samples here, or maybe elsewhere...)

Simulation (do we need this????)

The program is really just a layout operations simulator that produces switch lists that we we can follow in order to move the model cars around to match what the program has calculated. It does much more than just create switch lists, as it can

Studies (do we need this also???)

Output

Each study run will produce a new, unique directory that is named after the study so that nothing is overwritten. The output from running a study consists of two parts. The new layout state is saved to files, and a variety of switch lists and other analysis reports are also written out to this new directory. It is possible to read back in the events and layout files so that further post-processing of the run can be made.

Currently, the format of the switch lists is quite primitive and requires to use of a spreadsheet to format and print, but this will be improved over time.

Switch Lists

Include some samples and words here, more than what was touched upon earlier.

Two formats now, but others can be added as needed.....

Road

Yard Switcher (Simpler name???)

Instrumentation (Analysis?)

Various forms of reports can be produced by analysing the events from a study. One of the most useful is a chart of the minimum, maximum, and average number of cars at a particular track over a number of days.

Sample

The routing of a particular car over time can also be interesting to see how well it moves around the layout rather than being stuck in one route.

Sample

If the layout operation is not in balance over the longer term, this will manifest itself with clumps of cars accumulating in places instead of moving smoothly over the layout. Effects such as these often only show up when a study is run for many days.

Sample

Program Design

The program is named ProCFS or just CFS for short. It is written entirely in plain Java so that it can be

run on all of the major operating system platforms, including Mac OS X, Linux, and Windows. All of the data input and output is in plain text so that files can easily be inspected and moved between platforms.

There are numerous automated unit tests that are run constantly to ensure that the results of the program do not change as code is cleaned up and new features are added.

The program started life as a Windows 7 project written in C#, but was then converted to Java after Microsoft started to lose its way and I started to see the light. Early Java development was done still on a Windows machine, but is now entirely done on a Mac. There were no changes required to move the code base from Windows to OS X. It also has been tested on Linux without needing any changes.

(DO WE NEED THIS??????) Modular Logic:

The program has been structured in a decoupled manner so that it will be easy to insert more complex logic into various parts where it is desirable to consider more information. For instance the selection of a particular shipment for a car is decoupled from the logic that controls which train will move that car. Each piece of logic can easily be improved to handle more details as needed without having any affect on the other portions. As an example, the current matching of possible shipments to available cars is somewhat crude in that a boxcar door type and width is not considered when matching to shipments, it just has to be a boxcar. This can be improved without affecting any of the logic that controls where and how the car gets moved, as it will only affect the car selection for a given shipment request.

Program Size Limits and Performance

The program has no internal limits to any of the data structures. With a large amount of data, obviously there will eventually be memory size and performance issues, but testing indicates that it should be able to handle very large layouts. Performance is very good, even though no explicit work has been done to optimize the code yet. It has been written in a clear, simple, straightforward manner, so as to maximize correctness and ease of debugging. Significant performance improvements are very likely with some optimization effort.

Data File Formats

The layout information consisting of tracks, cars, shipments, waybills, etc. is stored in either one XML file or multiple related plain text CSV (comma separated value) files.

The intention of supporting CSV files is that data can be created and edited using simple spreadsheet programs using a columnar format instead of needing to use a program.

Study files are stored in XML format.

At the moment there is no editing capability within the program, so these files must be edited manually using a text editor or spreadsheet program. Eventually there will be full editing support so that the individual files do not need to be edited.

Data Import Capabilities

There is currently no ability to import data from other file formats, but this can be added as needed.

Current State

The majority of the core logic including car routing, shipment selection, and train movement has been written, thoroughly tested, and used by two model railroads for many operating sessions over a couple of years. The code has been continuously refined, and many new features added as the result

of numerous suggestions stemming from these sessions

What has yet to be written is a good user interface for entering and editing the data, and for running the simulations. The early emphasis was on "getting the answer right" instead of making the program pretty or easy to use. That can all come after it works correctly. Also, for expediency in proving an idea, some aspects of the program logic were hard coded for a particular layout rather than being controlled by user supplied data, and work has been underway to eliminate these shortcomings without affecting the end results.

Drawbacks

- Like any computer based system, the results are very prescriptive, in that everything is pre-calculated by the program, with the expectation that all of the moves will be carried out exactly by the human operators. This makes it hard to respond to unforeseen events during an operating session, such as a train that is late in arriving with cars that are needed by another train.

When considered on balance, the benefits of no manual set-up far outweigh the drawbacks of being somewhat prescriptive, in my opinion. Your mileage may vary and you will have to make this decision based on your own situation, wants, and dislikes.

Next Steps

Finish some clean-up work on the current code base.

Build an easy to use GUI editor for both the layout information and the studies.

Improve the charting capabilities of the program so that an external spreadsheet is not required.

A long term goal is a graphical editor for the layout track topology.

Explore an idea of having more flexible timing for some trains, such as a yard switcher, so that the expected time to complete each phase is more dependent on the actual amount of work to be done. This would provide a more realistic simulation, but in the end it is still a pre-calculated simulation with the same drawbacks.

S&BC Samples Online

All of the data files, plus sample switch lists, various reports, and charts will be available on my web site at http://sbcrailway.ca

Contact

Gregory Madsen

sbcrailway.ca