Passenger Rebooking Decision Modeling Challenge

Corticon Solution

Mike Parish

Flight	From	To	Dep	Arr	Capacity	Status
UA123	SFO	SNA	1/1/07 6:00 PM	1/1/07 7:00 PM	5	cancelled
UA456	SFO	SNA	1/1/07 7:00 PM	1/1/07 8:00 PM	2	scheduled
UA789	SFO	SNA	1/1/07 9:00 PM	1/1/07 11:00 PM	2	scheduled
UA1001	SFO	SNA	1/1/07 11:00 PM	1/2/07 5:00 AM	0	scheduled
UA1111	SFO	LAX	1/1/07 11:00 PM	1/2/07 5:00 AM	2	scheduled

Name	Status	Miles	Flight
Jenny	gold	500000	UA123
Harry	gold	100000	UA123
Igor	gold	50000	UA123
Dick	silver	100	UA123
Tom	bronze	10	UA123

RULES

1. Alternate flight must depart from the same place as the cancelled flight
2. Alternate flight must arrive at the same place as the cancelled flight
3. Alternate flight must depart after the cancelled flight
4. There must be room on the alternate flight
5. Passenger status determines who gets allocated first

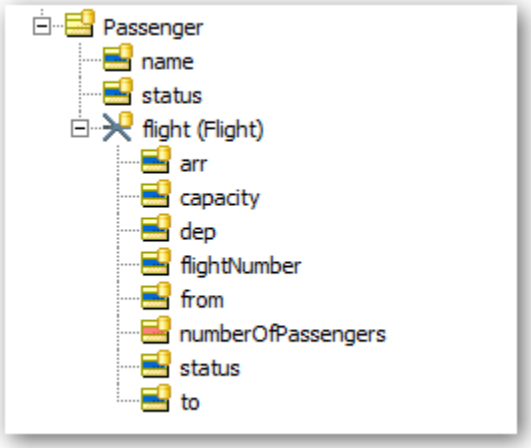
Results

Jenny is confirmed on UA456 departing SFO at 1/1/2007 19:00:00 arriving SNA at 1/1/2007 20:00:00
Harry is confirmed on UA456 departing SFO at 1/1/2007 19:00:00 arriving SNA at 1/1/2007 20:00:00
Igor is confirmed on UA789 departing SFO at 1/1/2007 21:00:00 arriving SNA at 1/1/2007 23:00:00
Dick is confirmed on UA789 departing SFO at 1/1/2007 21:00:00 arriving SNA at 1/1/2007 23:00:00
Tom could not be rebooked

Notes:

1. UA101 has zero capacity
2. UA456 and UA789 only have two seats each (these should go to the gold/silver passengers)
3. UA1111 has seats but only goes to LAX (so Tom is out of luck)

Vocabulary (Data Model)



The vocabulary shows a many to many relationship between passengers and flights.

A Basic Decision Table Solution

Natural Language View

Conditions	1	2	3	4
Is there an alternative flight that goes to the passengers final destination?	T	T	T	T
Does this alternative flight depart from the passengers origin airport	T	T	T	T
Does this alternative flight depart after the cancelled flight departure time?	T	T	T	T
Is there room on this alternative flight?	T	T	T	T
What is the status of the passenger?	'gold'	'silver'	'bronze'	other
Actions				
Post Message(s)	✉	✉	✉	✉
Take the passenger off the cancelled flight	☑	☑	☑	☑
Put the passenger on the alternative flight	☑	☑	☑	☑

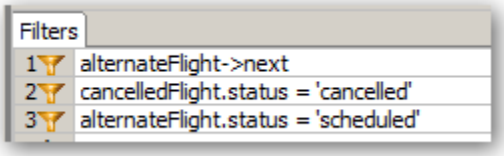
Implementation View

Conditions	1	2	3	4
cancelledFlight.to = alternateFlight.to	T	T	T	T
cancelledFlight.from = alternateFlight.from	T	T	T	T
cancelledFlight.dep <= alternateFlight.dep	T	T	T	T
alternateFlight.capacity > alternatePassengerList->size	T	T	T	T
inconveniencedPassenger.status	'gold'	'silver'	'bronze'	other
Actions				
Post Message(s)	✉	✉	✉	✉
cancelledFlight.passengers-=inconveniencedPassenger	☑	☑	☑	☑
alternateFlight.passengers+=inconveniencedPassenger	☑	☑	☑	☑

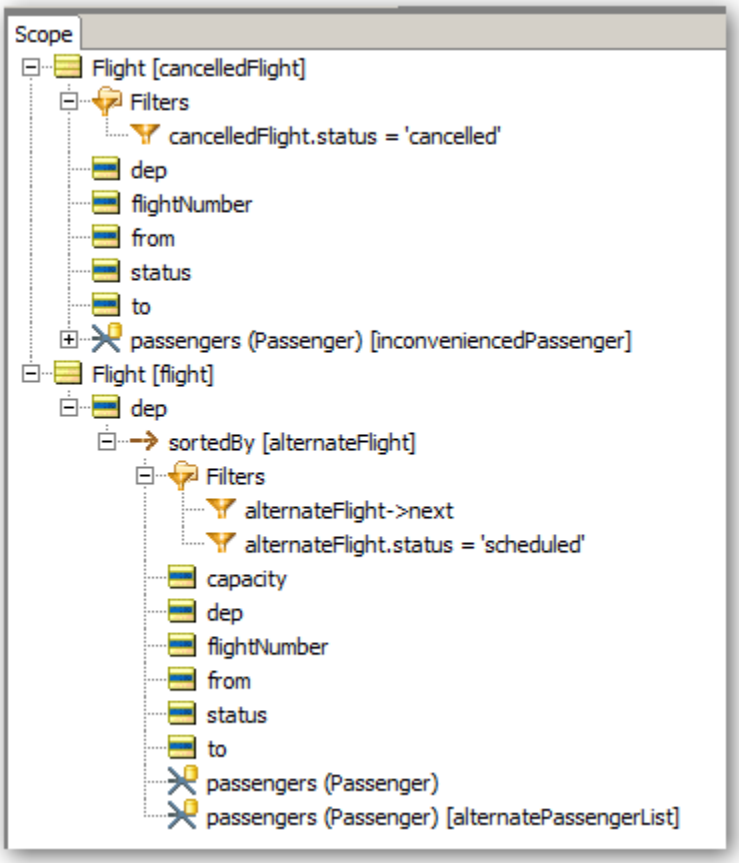
This arrangement causes Corticon to consider all the gold passengers first and then the silver and then the bronze and finally anyone else. This structure is sometimes useful when the statuses are not easily sorted.

Filters

These define a cancelled flight and an alternative flight.



Filter 1 simply takes the alternative flights in a particular order which is defined in the scope using sortedBy:



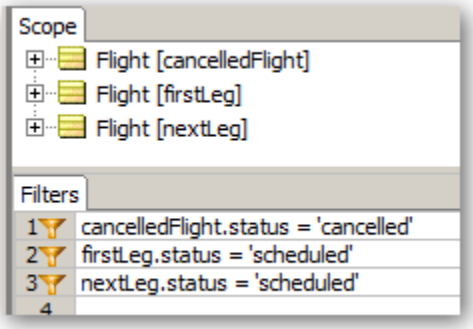
Currently the alternative flights are sorted by ascending departure time so passengers get the first available flight out. However that may not necessarily be the quickest way to their final destination. Perhaps customer would prefer to be assigned to flights that arrive at their final destination earlier even if it means departing later. To make this rule change we simply sort on the arrival time of the alternative flight.

You can also see in the scope how we have used different aliases to refer to the different groupings of passengers. When they are on the cancelled flight they are referred to as inconveniencedPassenger but when they are on a scheduled flight they are referred to as alternatePassengerList. The rules move passengers out of one group into the other

Enhancement - Indirect Flights

Indirect flights (having one stop) can be handled with a minor change to the rules

Conditions	0	1	2	3	4
cancelledFlight.to = nextLeg.to		T	T	T	T
cancelledFlight.from = firstLeg.from		T	T	T	T
firstLeg.to =nextLeg.from		T	T	T	T
cancelledFlight.dep <= firstLeg.dep		T	T	T	T
nextLeg.dep > firstLeg.arr		T	T	T	T
firstLeg.capacity > firstLegPassengerList->size		T	T	T	T
nextLeg.capacity > nextLegPassengerList->size		T	T	T	T
inconveniencedPassenger.status		'gold'	'silver'	'bronze'	other
Actions					
Post Message(s)		✉	✉	✉	✉
cancelledFlight.passengers-=inconveniencedPassenger		☑	☑	☑	☑
firstLeg.passengers+=inconveniencedPassenger		☑	☑	☑	☑
nextLeg.passengers+=inconveniencedPassenger		☑	☑	☑	☑



Other enhancements might include a rule sheet to determine the passenger priority based on their status and number of miles traveled.

Summary

Even quite complex problems involving multiple associated objects can be solved easily using decision tables in Corticon.

This is because it is able to handle collections of objects (represented by 1:N or N:N associations) using its scope (to declare different collection aliases) and filter section (to define the membership of those collections).

The rules can then refer to those collection aliases.