Feature Description:

Enhanced Agent Requirements Forecasting

For Community Software version 3.x – 4.x

Revision 1.2

July 22, 2021

CommunityWFM

phone 877-668-6870
web CommunityWFM.com

3400 Waterview Parkway, Suite 101
Richardson, Texas 75080

©2021, CommunityWFM. All Rights Reserved.

CommunityWFM

Table of Contents

About this Document 3

Why Enhanced Agent Requirements Forecasting? 4

How does Community Generate the Clusters? 6

What do the Clusters do? 7

Forecasting for Dissimilar Service Objectives 9

Rollup Values for Parent Activities / Calculating Staffing Hours / Calculating FTEs 10

Scheduling Standard Erlang vs. 10

Enhanced Agent Requirements Forecasts 10

phone 877-668-6870
web CommunityWFM.com

3400 Waterview Parkway, Suite 101
Richardson, Texas 75080

About this Document

The purpose of this document is to describe in detail the Enhanced Agent Requirements Forecasting module, available in Community Release 3.x and greater. With this feature, Community can provide significant reduction in the required headcount based on agent cross-training and thereby provide a more accurate forecast of the number of agents required to satisfy the contact center load.

Why Enhanced Agent Requirements Forecasting?

Community, like most current workforce management applications, provides agent requirements forecasting based on Erlang-C mathematical formulas to convert calls, handle times, and service objectives into agents required. While the Erlang-C formulas have proven over time to be reliable in their calculations, they do not account for multiskilled agents in the contact center. The formula assumes that agents will be dedicated to servicing only the demand from a single skill.

The concept behind the "clustering" strategy is to group activities together for the purposes of generating agent headcount requirements by analyzing the cross-training matrix for the agents involved in a schedule. As the cross-training increases, the number of agents required to satisfy the demand in a single activity is reduced because, again, agents are shared across multiple activities. The clustering should be performed such that the allocation is proportional to the call volume for the activity, but relative to the number of agents that also have the particular activity assignment.

Once the activities are clustered, they can be forecasted as a single activity and the agent cross training efficiencies are realized. The cluster represents a group of related activities, and for member activity in a cluster, Community calculates a probability of a call being answered by the given activity in the given cluster. The more disjointed the cross-training matrix for the agents in the schedule, the more clusters will result.

Consider a typical example.
(For all examples presented in this document, assume an 80/20 service level.)

Activity Name	Contacts	Average Handle Time	Agents Required
English	100	250	33
Spanish	50	300	21
French	25	320	13
Total for Time Interval			**67**

First, assume that no cross training exists between agents assigned to the above skills. This would appear as in the following agent assignment table:

Agent	English	Spanish	French
Agent1	X		
Agent2		X	
Agent3			X

In this case, the Erlang-C based agent requirements forecast is perfectly correct, because no efficiencies can be realized from agent cross-training.

Now, consider the other extreme, where each agent is assigned all available skills. In this case, the agent assignment table appears as follows:

Agent	English	Spanish	French
Agent1	X	X	X
Agent2	X	X	X
Agent3	X	X	X

The fundamental principle behind the clustering algorithm is that the demand (contact volume and handle time) for commonly assigned skills can be combined into a single skill and re-forecasted using the standard Erlang-C formula.

In the above example, all agents are available in all skills, so essentially the demand can be represented as a single composite skill. In this case, the following will be forecasted:

Activity Name	Contacts	Weighted Handle Time	Agents Required
Combined	175	274	**59**

In this example, we can see that the cross training has provided a 12% agent efficiency gain (8 agents) simply by combining the skills cross trained. Note the Weighted Handle Time column is the volume-weighted average handle time calculated as:

SUM(Average Handle Time * (SUM(Volume)))

In this example:

(250 * (100 / 175)) + (300 * (50 / 175)) + (320 * (25 / 175)) = 274 wAHT

However, in most circumstances the agent population is neither 100% cross-trained nor 0% cross-trained. The result is that the ability to group commonly assigned activities is significantly difficult. To accomplish this, Community uses an algorithm based on a quadratic equation to determine the similarity in skills across agents. The result is a set of activity clusters and the probability that a call will be answered by a given activity in each cluster.

How does Community Generate the Clusters?

The clusters provide the probability that a call can be answered in a skill in a cluster. In the fully cross-trained example above, Community would create a single cluster that represents all skills and agents. The result is that each skill (activity) in the cluster would have a 100% probability of being satisfied in that cluster, simply because no other clusters exist.

However, if cross training is fragmented, more clusters result, and the probabilities will vary depending on the clusters involved. Consider a more realistic cross training matrix:

Agent	English	Spanish	French
Agent1	X		X
Agent2	X	X	
Agent3	X	X	X

During the process of creating an Agent Requirements Forecast, Community generates the clusters based on the above matrix. The result is three clusters for each activity, due to the fragmentation of the French and Spanish activities. The resulting cluster matrix appears as follows:

Activity	Cluster 0	Cluster 1	Cluster 2
English	33.33	33.33	33.33
Spanish	50.00	50.00	0.00
French	0.00	50.00	50.00

Examining the clusters will reveal that Cluster 0 is the combination of the English and the Spanish activities, Cluster 1 is the combination of all three activities, and Cluster 2 is the combination of the English and the French activities.

Cluster 0 is a result of the cross training provided by Agent2, who is assigned only English and Spanish. Cluster 1 is a result of the cross training provided by Agent3, who is assigned all three activities. Cluster 2 is a result of the cross training provided by Agent1, who is assigned only English and French.

Notice also that the English activity appears in all clusters. The reason is because all agents are assigned to the English activity, so in all clusters some chance exists that a call offered to the English activity will be answered in each cluster. Since all agents are assigned English, the probability is evenly divided across the clusters.

What do the Clusters do?

Again, the clusters provide the probability that a call offered to an activity will be handed by the activity in the cluster. The probabilities for an activity in the clusters will always sum to 100 (100% probability that the call will be answered by the activity in a cluster). Essentially, for a given cross training matrix, the clusters provide the right combinations of activities to forecast as a single combined group. The correct grouping of activities exists somewhere between the two extremes (no cross training and 100% cross training), and the clusters provide the grouping strategy.

The first step in creating the enhanced forecast is to determine the number of servers given the volume and handle time for all activities included in the cluster. Essentially, the algorithm pools all the volume for all activities together, as in the 100% cross training example above. However, the volume for activities is not included in the pooled demand if the probability of a call being handled in that cluster is 0. The following table illustrates the pooled demand resulting from the clusters defined above:

Cluster	Pooled Contacts	Pooled WAHT	Servers (Agents)
Cluster 0	150	267	50
Cluster 1	175	274	59
Cluster 2	125	264	42

Recall that Cluster 0 is a combination of English and Spanish, which contribute 100 and 50 calls to the combined "pool" respectively. The Weighted Average Handle Time is calculated as described above.

Cluster 1 is a combination of all three activities, resulting in a pooled demand of 175 (100 + 50 + 25), and Cluster 2 is a combination of English and French with pooled demand of 125 (100 + 25).

The algorithm then calculates the number of servers for the entire combined demand using the service objectives of the current activity. The algorithm will sum the pro-rated servers across all clusters where the probability is > 0. The formula below provides the clustered servers for a given activity:

Clustered Servers = SUM(wC * sP * (cA / cP))

Where:

- wC is the weight for a given cluster
- sP is the servers for the pooled demand
- cA is the volume for the given activity
- cP is the volume for the given pool
- Substituting our values for the above activities:

English

	wC	sP	cA	cP	Agents
Cluster 0	.3333	50	100	150	11
Cluster 1	.3333	59	100	175	12
Cluster 2	.3333	42	100	125	11
Total					**34**

Spanish

	wC	sP	cA	cP	Agents
Cluster 0	.5	50	50	150	8
Cluster 1	.5	59	50	175	9
Cluster 2	0	0	0	0	0
Total					**17**

French

	wC	sP	cA	cP	Agents
Cluster 0	0	0	0	0	0
Cluster 1	.5	59	25	175	4
Cluster 2	.5	42	25	125	4
Total					**8**

The accumulation of all agent requirements is 34 + 17 + 8, or **59** agents. Compared to the original Erlang C based agent requirements forecast of 67, we realize a 10 agent efficiency gain, or approximately 15% for this particular interval. Note: all values for agents required are rounded up to the nearest whole number to ensure the desired service levels are achieved.

The efficiency calculation to compare the Standard Forecast versus the Enhanced Forecast is as follows:

Staffing Hours – Enhanced Staffing Hours / Staffing Hours

> **Note**: View the cluster weights in Community with the link "…review cross-training cluster reports" (in a working forecast, go to Staffing requirements and select the Enhanced forecasting tab). However, the activity volume and pooled volume are not directly exposed in the application. These can be manually calculated by adding the

Forecasting for Dissimilar Service Objectives

The above algorithm calculates the headcount required to satisfy a given activity's proportion of the pooled demand based on the activity's service objectives. While the volume and handle time values for the pool is used across multiple activities, the servers (agents) required is based on the specific activity's service objectives. Therefore, the resulting headcount requirement for the pool is based on the service level for the activity and consequently the prorated amount reflects the individual service objective.

Consider the above example with a service level for Spanish at 90/10.

Spanish

	wC	sP	cA	cP	Agents
Cluster 0	.5	54	50	150	9
Cluster 1	.5	63	50	175	9
Cluster 2	0	0	0	0	0
Total					**18**

The resulting headcount increases for Cluster 0, reflecting the increased service level required for Spanish. However, other activities are unaffected by the dissimilar service objective.

Rollup Values for Parent Activities / Calculating Staffing Hours / Calculating FTEs

The method for displaying agent requirements at a parent activity level is the same between Standard Forecasting and Enhanced Forecasting. The staffing hours at any level of the activity are calculated as follows:

SUM(Interval Head Count Requirements / System Default Slots per Hour)

In the examples above, if each interval between 7:00 AM and 5:00 PM (inclusive) required 59 agents, then the staffing hours at the parent level would be:

Agents Required * Slots/Hour * Hours (59 * 4 * 10) + 59 = 604 Staffing Hours

Note the last 59 is due to the 5:00 PM interval being included in the forecast.

Calculating the FTEs based on the staffing hours is:

Staffing Hours / 8.0 (Assume FTE is 8-hour employee).

These values for an individual activity are identical to a parent activity.

Scheduling Standard Erlang vs. Enhanced Agent Requirements Forecasts

Community provides schedules the ability to choose the forecasting method used to generate schedules. If the Enhanced Agent Requirements Forecast is selected (using Skill-based Forecasting Methods), Community exports the Enhanced Agent Requirements value for the interval. If standard forecasting is selected, then Community exports the standard Erlang based agent forecast. However, in either case the optimizing engine will optimize the workforce over all assigned activities.