1. Distribution Rule specification

Distribution Rule in DCM is used by ACD to route a contact to an eligible agent. The eligible agent must match **the skill selector**, defined in Distribution Rule. And in high level, distribution rule can be a single **rule**, a **rule group**, or **rule set**.

* 1. Skill Selector

Skill Selector is the criteria to determine if the agent is eligible to handle the contact. It can be:

* **Single skill selector**: the selector has only one skill with or without level condition. Here is the syntax:

***@S***: {skill} [ ***level*** (

(***<|<=|=|<>|>=|>***) {number}) | (***in*** {number} .. {number})

)]

Where {skill} is skill oid, and {number} is any positive integer.

* **Skill set selector:** a set of single skill selectors, and the relationship amount those single skill selectors is AND condition. Here is the syntax:  
  {***single skill selector***}, {***single skill selector}+***

Here are some examples:

Ex 1: *@S: 1111a level < 3*  
 If agent has skill 1111a and skill level is greater than 3, then the agent can be in this skill queue target.

Ex 2: *@S: 1111a level in 3 .. 5*

If agent has skill 1111a and skill level is between 3 to 5, then the agent can be in this skill queue target.

Ex 3: *(@S: 1111a level = 3, @S: 2222a level > 1)*

If agent has skill 1111a with level 3 and skill 2222a with level is greater than 1, then the agent can be in this skill queue target.

* 1. Rule

A single routing rule that can be executed by ACD to find eligible agents. It starts with “**queue to**” key word and ends with the number priority.

***queue to*** *[****least busy of****] {skill selector} [and {skill selector}] with priority {number}*

where:

* the relationship amount skill selectors are OR condition (Event the rule has “and”).
* {number} is a position number to indicate the priority of the rule. The priority will be used in Rule Group, or Rule Set.
* ***least busy of:*** indicates the rule should find the least EWT-CT of skill selector, and then get agent from the skill selector.

**EWT-CT**: the estimated wait time that a contact could expect to wait before an agent answers it if it is routed to the specified CT. The estimated wait time for the CT is calculated using the following information:

* the number of contacts currently in the skill queue ahead of the contact at the specified priority.
* the number of contacts currently int queue ahead of the contact at a higher priority.
* the number of agents currently logged in who possess any skill to which the contact is currently queued.
* the average handle time (AHT) of the current contact type.

If at least one agent is available to handle a contact of the specified type, then the EWT-CT will be zero.

If there are no logged-in agents who can handle the specified CT, the EWT-CT will be infinite.

If the only CT specified in the EWT-CT variable is invalid, then EWT-CT will be infinite number.

Here are some examples:

Ex 1: ***queue to @S:1111a with priority 1***

With this rule, the contact will be assigned to any agent that has skill 1111a.

Ex 2: ***queue to @S:1111a level < 10 with priority 1***

With this rule, the contact will be assigned to any agent that has skill 1111a and level is less than 10.

Ex 3: ***queue to @S:1111a and @S:2222a with priority 1***

With this rule, the contact will be assigned to any agent that has skill 1111a, OR skill 2222a.

Ex 4:

***queue to (@S:1111a, @S:2222a) with priority 1***

With this rule, the contact will be assigned to any agent that has skill 1111a and skill 2222a.

Ex 5: ***queue to (@S:1111a, @S:2222a) and @S:3333a with priority 1***

With this rule, the contact will be assigned to any agent that has skill 1111a and skill 2222a, or agent that has skill 3333a.

Ex 6: ***queue to least busy of @S:1111a and @S:2222a and (@S:1111a, @S:3333a) with priority 1***

This rule contains **“least busy of”**, it means the rule should find the least EWT of 1111a skill queue, 2222a skill queue, or (1111a skill and 3333a skill). And if the least EWT is skill queue (1111a skill and 3333a skill), then the contact will be assigned to any agent that has skill 1111a and skill 3333a.

* 1. Rule Group

Rule Group consists with more than one Rue, the priority in each rule must be unique, and the relationship between Rules is OR condition. Here is the syntax:

***Rule [Rule]***

For example:

***queue to @S: skillOne and @S: skillTwo with priority 1***

***queue to @S: skillThree and @S: skillFour with priority 2***

with this rule, ACD is looking for agent with skillOne, or agent with skillTwo first. If no found, looking for agent with skillThree, or agent with skillFour.

* 1. Rule Set

Rule set consists with more Rule Groups with waiting time between each Rule Group. Here is the syntax:

***Rule Group [wait*** *{number}* ***Rule Group]***

Where {number} is the positive number, indicates wait time in seconds.

For example:

***queue to @S: skillOne and @S: skillTwo with priority 1***

***queue to @S: skillThree and @S: skillFour with priority 2***

***wait 300***

***queue to @S: skillFive with priority 3***

This rule means: after the contact is arriving and waiting less than 300 seconds, the rule is

***queue to @S: skillOne and @S: skillTwo with priority 1***

***queue to @S: skillThree and @S: skillFour with priority 2***

after waiting more than 300 seconds, the rule is

***queue to @S: skillOne and @S: skillTwo with priority 1***

***queue to @S: skillThree and @S: skillFour with priority 2***

***queue to @S: skillFive with priority 3***

1. Distribution Rule Script to Java Object

Distribution Rule is script language in Backus Normal Form. In java, ANTLR (Another Tool for Language Recognition) can be used to parse the script and we have two options:

* Convert the script language into java objects and ACD run those rule objects.
* Create script runtime engine and execute the script when ACD is doing contact assignment.

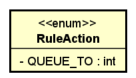
The first approach has the following advantages:

* The script is parsed only once, which has better performance.
* Decouple the script and ACD action, easy to test.

We will try to take the first approach unless the rule become too complicated and not possible to convert into java object.

1. Rule Classes

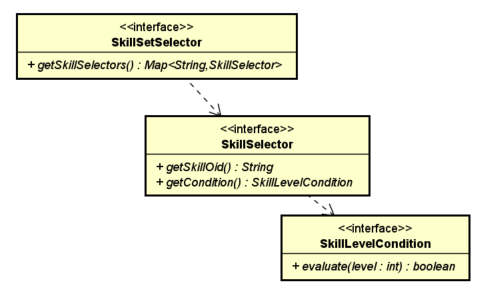
First, the current distribution rule only supports “queue to” action but it may support other action. We define enum class – RuleAction to represent the action of rule but it only contains QUEUE\_TO.



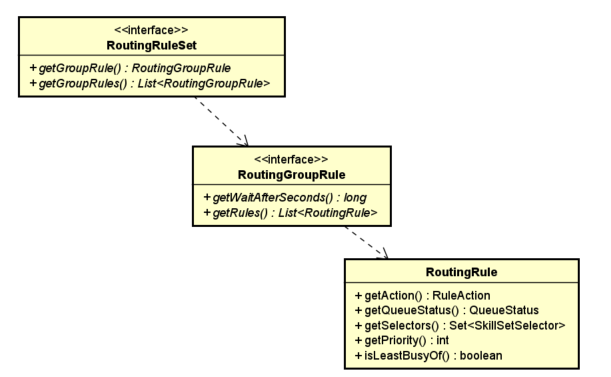
Second, “least busy of” is used to select the least EWT skill queue. We define enum class – QueueStatus to represent this.



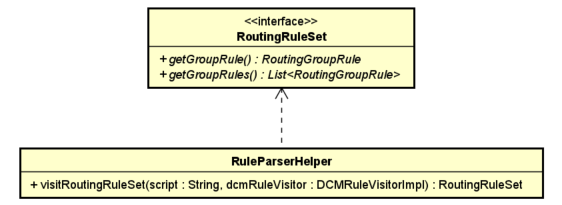
Third, the skill queue has two options: a single skill with binary condition or sql like condition, a set of a single skill selectors. For this part, we define three interfaces:



And we define three interfaces to represent Rule, Rule Group, and Rule Set:

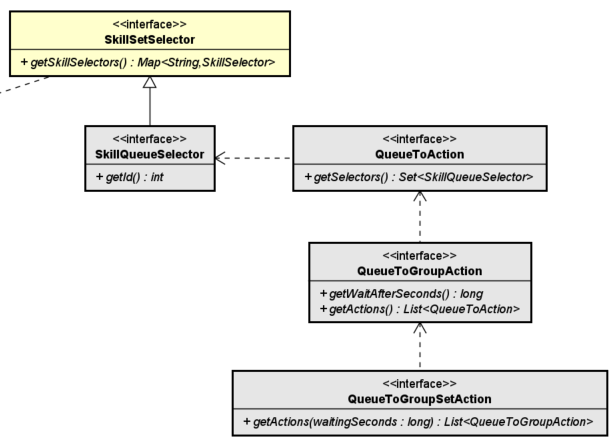


1. Parser Helper  
   A Parser Helper class – RuleParserHelper can be used to parse rule script and return the implementation class of RoutingRuleSet.



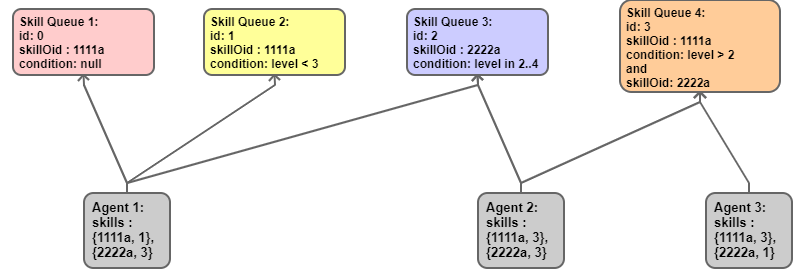
1. Distribution Rule Service

In future, the rule may extend to add some if-then-else logic, or some other unpredictable rule syntax. The action layer is added as wrapper on top of rule classes. And this layer may help use to keep the backward compatibility.

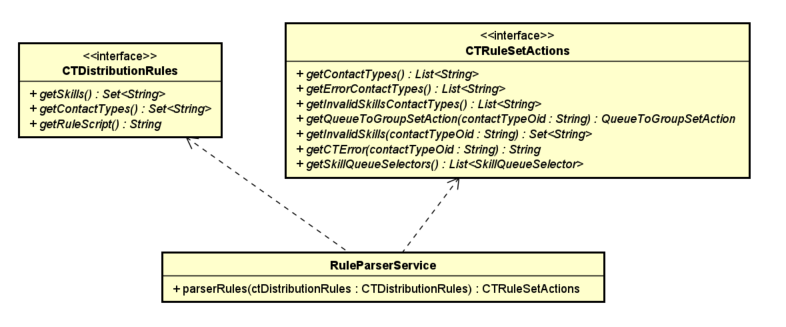


The interface of SkillSetSelector defines a set of skills with conditions, and it is used to select the eligible agent. In the action layer, SkillQueueSelector is extended from SkillSetSelector but add a unique id. And ACD can use SkillQueueSelector to create skill queue, and eligible agent can register to skill queues.

For example, four queue selectors are from all contact distribution rules. And after agents is online, or available, or skill set is changed, agents can register to skill queue that matches agent’s skill set.



And the service, RuleParserService, can be used to parse distribution rules of all contact types:



And follow the java standard service locator framework, RuleParserServiceProvider is added to be used to create RuleParserService.

