



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

SMART User Manual

Program Triggers

5/3/2025

Program Triggers

Program Triggers define the circumstances under which SpeechMiner should search for a dependent topic. They are defined in the **program structure** and are represented by arrows that connect the topics.

The following expressions and conditions can be used when creating a Program Trigger:

Logical Expressions

Configuring Triggers is performed by defining a logical expression for SpeechMiner to evaluate. If the logical expression is evaluated as True, the Trigger is activated and the child **recognition task** is run on the interaction. If the logical expression is evaluated as False, the Trigger is not activated, and the child recognition task is not run.

Logical expressions enables you to configure the relationship between the topic and its dependent topic. For example, you can specify that a topic must be recognized with a particular level of strictness or the topic must be found twice before SpeechMiner searches for its dependent topic.

Logical expressions can consist of one or more elements. Each element is a statement, such as Date Of Interaction>1/19/2010 or Found topic "Reduction" at least once with Low Strictness. If a logical expression contains one element, it is True whenever the condition defined in the element is true. For example, if a logical expression contains the element Date Of Interaction>1/19/2010, it is True for any interaction whose metadata indicates that it took place on or after 1/20/2010.

Logical expressions can also contain multiple elements. When they do, the elements must be related to one another by logical operators, such as AND or OR. For example, Date Of Interaction>1/19/2010 AND Found topic "Reduction" at least once with Low Strictness is a logical expression that indicates both conditions must be met (the topic must be found and the interaction must have taken place in the specified time period), in order for the expression to be evaluated as True and the recognition task to be implemented.

The following operators can be used in logical expressions:

Operator	Description
AND	Both conditions must be met in order for the expression to be evaluated as True.
OR	At least one of the conditions must be met in order for the expression to be evaluated as True.
NOT	The expression should be evaluated as True only when the condition is not met.
(Opens a group of conditions that are evaluated as a unit.
)	Closes a group of conditions that are evaluated as a unit.

Logical expressions can be quite complex, containing a number of elements, various logical operators, and parentheses, if necessary. For example, the following logical expression means that the recognition task should be performed if the interaction took place between Jan. 1st and Jan. 30th and the topic "Reduction" was found, or if the interaction took place Jan. 31st or later and either the topic "Time Frame" or the topic "Payment Locations" was found.:

(Date Of Interaction>1/1/2011 AND Date Of Interaction<1/30/2011 AND Found topic "Reduction" at least once with Low Strictness) OR (Date Of Interaction>1/31/2011 AND (Found topic "Time Frame" at least once with Low Strictness OR Found topic "Payment Locations" at least once with Medium Strictness))

Related Topics

[Create a Program Trigger](#)
[Recognition Tasks](#)

Parentheses in Expressions

Parentheses can be inserted into an expression to group elements together for evaluation as a single unit. Complex expressions often require parentheses to break them into smaller expressions so that the relationships between the expressions are clear. For example, if a topic must be found in interactions handled by one of two agents, and you write the following expression, it is not clear whether the topic must be found in interactions handled by both agents or only in interactions handled by agent b.

Workgroup\Agent=workgroup 1/agent a OR Workgroup\Agent=workgroup 1/agent b AND Found topic "Reduction" at least once with Low Strictness

If you add parentheses as follows, the topic should be found in interactions handled by one of the agents:

(Workgroup\Agent=workgroup 1/agent a OR Workgroup\Agent=workgroup 1/agent b) AND Found topic "Reduction" at least once with Low Strictness

If you write the following expression, the topic is only be required in interactions handled by agent b:

Workgroup\Agent=workgroup 1/agent a OR (Workgroup\Agent=workgroup 1/agent b AND Found topic "Reduction" at least once with Low Strictness)

Related Topics

[Create a Program Trigger](#)
[Recognition Tasks](#)

Metadata Conditions

Metadata refers to the interaction data gathered by the external recording system and relayed to SpeechMiner along with the interaction. Each system receives different sets of information about the interactions that are passed to it for processing. For example, metadata could include the name of the agent for each interaction, the location of the customer, and the duration of the interaction.

Metadata Conditions enables you to instruct SpeechMiner to search for a specific dependent topic when the interaction being analyzed meets one or more metadata requirements. For example, you can specify that SpeechMiner should only search for a specific topic when the customer is in the state of Nevada, or if a particular agent handled the interaction.

You can define metadata conditions based on any of the metadata values available in your system. In addition, you can define different metadata conditions for each topic in a program. The conditions you define can be as simple as one metadata value or complex expressions that specify the relationships among a number of metadata types. Take, for example, a sales campaign that is undertaken in Wyoming and in Texas. Certain agents in each location target senior citizens, while others target families with teenage children. To process interactions to senior citizens differently from other interactions that belong to the program, you could define a condition that specifies either Agent1 or Agent2 if the customer is in Wyoming, and Agent3 if the customer is in Texas, and apply the condition to the topics that are specific to the senior-citizen campaign.

Related Topics

[Create a Program Trigger](#)
[Recognition Tasks](#)

Negative Condition

A negative condition is a condition that must not be met if an expression is to be evaluated as True. For example, a condition that states that an interaction must not have taken place during the month of January is a negative condition. Negative conditions can be specified for a single element of an expression or a group of elements that are contained in parentheses.

Two methods are available for creating a negative condition for a single element:

- Insert a NOT operator before the element.
- Select the Not option in the element's conditions.

The following are two examples of the same negative condition written in two different ways. These

conditions are True as long as the date is not on or before January 31st, 2011.

NOT Date Of Interaction<1/31/2011

- Or -

Not Date Of Interaction

An expression containing multiple elements can only be marked as a negative condition using the **NOT** operator. For example, in the following expression, the NOT operator is attached to a multi-element condition. The condition is True for all workgroups except workgroup 1 and workgroup 4.

NOT (Workgroup\Agent=/workgroup 1 OR Workgroup\Agent=/workgroup4)

Related Topics

[Create a Program Trigger](#)
[Recognition Tasks](#)