



ARL-TN-1172 • SEP 2023



# Inferencing Model Documentation

by Somiya Metu and Adrienne Raglin

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



# Inferencing Model Documentation

**Somiya Metu and Adrienne Raglin**  
*DEVCOM Army Research Laboratory*

## REPORT DOCUMENTATION PAGE

<b>1. REPORT DATE</b>		<b>2. REPORT TYPE</b>		<b>3. DATES COVERED</b>	
September 2023		Technical Note		<b>START DATE</b>	<b>END DATE</b>
				12/01/22	03/01/23
<b>4. TITLE AND SUBTITLE</b>					
Inferencing Model Documentation					
<b>5a. CONTRACT NUMBER</b>		<b>5b. GRANT NUMBER</b>		<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>5d. PROJECT NUMBER</b>		<b>5e. TASK NUMBER</b>		<b>5f. WORK UNIT NUMBER</b>	
<b>6. AUTHOR(S)</b>					
Somiya Metu and Adrienne Raglin					
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-IC Adelphi, MD 20783				ARL-TN-1172	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>			<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b>					
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b>					
This report documents an inferencing model that has been developed using the Sentry Agents (SAGE) framework. The documentation herein describes SAGE-based agent behaviors that were developed to perform basic inference.					
<b>15. SUBJECT TERMS</b>					
Military Information Sciences, artificial intelligence, inferencing, agent-based systems, Sentry Agents, SAGE					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>		<b>18. NUMBER OF PAGES</b>
<b>a. REPORT</b>	<b>b. ABSTRACT</b>	<b>c. THIS PAGE</b>	UU		16
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED			
<b>19a. NAME OF RESPONSIBLE PERSON</b>				<b>19b. PHONE NUMBER (Include area code)</b>	
Somiya Metu				(301) 394-1398	

**STANDARD FORM 298 (REV. 5/2020)**

*Prescribed by ANSI Std. Z39.18*

## Contents

---

<b>List of Tables</b>	<b>iv</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Inference Model Classes</b>	<b>1</b>
2.1 Class SensorInfoUoC	1
2.2 Class QueryToInfer	3
2.3 Class SensorDatastore	5
<b>3. Conclusion</b>	<b>7</b>
<b>4. References</b>	<b>8</b>
<b>List of Symbols, Acronyms, and Abbreviations</b>	<b>9</b>
<b>Distribution List</b>	<b>10</b>

## List of Tables

---

Table 1	Class SensorInfoUoC method summary .....	1
Table 2	Class SensorInfoUoC field summary.....	3
Table 3	Class QueryToInfer method summary .....	3
Table 4	Class QueryToInfer field summary .....	4
Table 5	Class SensorDataStore method summary .....	5
Table 6	Class SensorDataStore field summary .....	7

## 1. Introduction

---

This report documents an inferencing model that has been developed using the Sentry Agents (SAGE<sup>1</sup>) framework. SAGE, an open-source software developed by the Naval Research Laboratory, can be used to create agent-based systems with agents that can have their own custom behaviors. The documentation herein describes SAGE-based agent behaviors that were developed to perform basic inference. The inferencing model utilizes sensor reading values from a sensor data set to infer and detect movements.

## 2. Inference Model Classes

---

This section describes the classes that are present in the inferencing model.

### 2.1 Class SensorInfoUoC

---

The SensorInfoUoC class describes the data structure that represents sensor data set entries (Tables 1 and 2).

**Table 1 Class SensorInfoUoC method summary**

<b>Modifier and type</b>	<b>Method and description</b>
Double	<b>getA()</b> Get sensor type A value <b>Returns:</b> sensor type A value
Double	<b>getB()</b> Get sensor type B value <b>Returns:</b> sensor type B value
Double	<b>getC()</b> Get sensor type C value <b>Returns:</b> sensor type C value
String	<b>getRowID()</b> Get unique entry identification <b>Returns:</b> unique entry ID
Double	<b>getTime()</b> Get sensor reading time
String	<b>getUoI_A_Sensor()</b> Get Uncertainty of Information <sup>2</sup> (UoI) value related to sensor A <b>Returns:</b> UoI value related to sensor type A

**Table 1 Class SensorInfoUoC method summary (continued)**

<b>Modifier and type</b>	<b>Method and description</b>
String	<b>getUoI_B_Sensor()</b> Get Uncertainty of Information (UoI) value related to sensor B <b>Returns:</b> UoI value related to sensor type A
String	<b>getUoI_C_Sensor()</b> Get Uncertainty of Information (UoI) value related to sensor C <b>Returns:</b> UoI value related to sensor type A
void	<b>printme()</b> Print sensor information
void	<b>setA(java.lang.Double tmpA)</b> Set sensor type A value <b>Parameters:</b> tmpA - sensor A value
void	<b>setB(java.lang.Double tmpB)</b> Set sensor type B value <b>Parameters:</b> tmpB - sensor B value
void	<b>setC(java.lang.Double tmpC)</b> Set sensor type C value <b>Parameters:</b> tmpC - sensor C value
void	<b>setrowID(java.lang.String d)</b> Set unique entry identification <b>Parameters:</b> d - unique entry ID
void	<b>setTime(java.lang.Double time)</b> Set sensor reading time <b>Parameters:</b> time - sensor reading time
void	<b>setUoI_A_Sensor(java.lang.String a)</b> Set Uncertainty of Information (UoI) value related to sensor type A <b>Parameters:</b> a - UoI value related to sensor type A
void	<b>setUoI_B_Sensor(java.lang.String b)</b> Set Uncertainty of Information (UoI) value related to sensor type B <b>Parameters:</b> b - UoI value related to sensor type B
void	<b>setUoI_C_Sensor(java.lang.String c)</b> Set Uncertainty of Information (UoI) value related to sensor type C <b>Parameters:</b> c - UoI value related to sensor type C

**Table 2 Class SensorInfoUoC field summary**

<b>Field name</b>	<b>Field detail</b>
Double	<b>A</b> Sensor type A value in the sensor data set
Double	<b>B</b> Sensor type B value in the sensor data set
Double	<b>C</b> Sensor type C value in the sensor data set
String	<b>rowID</b> Unique entry identification
Double	<b>Time</b> Sensor reading time stamp
String	<b>UoI_A_Sensor</b> Uncertainty of Information (UoI) value related to sensor type A
String	<b>UoI_B_Sensor</b> Uncertainty of Information (UoI) value related to sensor type B
String	<b>UoI_C_Sensor</b> Uncertainty of Information (UoI) value related to sensor type C

## 2.2 Class QueryToInfer

The QueryToInfer class represents a SAGE behavior that sends intermittent queries to an agent associated with SensorDataStore behavior (Tables 3 and 4). The result of the query is then processed to infer the detection of movements utilizing sensor readings.

**Table 3 Class QueryToInfer method summary**

<b>Modifier and type</b>	<b>Method and description</b>
boolean	<b>action</b> (nrl.sage.BehaviorInterface.Result result) This method is called intermittently by the agent after the agent becomes active. It initiates a call to the query method. <b>Parameters:</b> result - SAGE result object to carry information about the result of method execution
void	<b>assembleSensorInfo</b> (java.lang.String str) <b>Parameters:</b> str - String representing a SensorInfoUoC type
	<b>infer</b> () This message utilizes the result HashMap that is created from the query that is sent to the agent with SensorDataStore behavior.
void	

**Table 3 Class QueryToInfer method summary (continued)**

<b>Modifier and type</b>	<b>Method and description</b>
boolean	<p><b>message</b>(nrl.sage.BehaviorInterface.Message message, nrl.sage.BehaviorInterface.Result result)                      This method is called each time a message is received by the agent.</p> <p><b>Parameters:</b>                      message - Java class that represents a message sent between SAGE agents for communication.                      result - Java class that represents the outcome of the execution of the message method.</p>
void	<p><b>query</b>()                      This method randomly pulls a string from the array holding query strings.</p>
boolean	<p><b>sendMessage</b>(java.lang.String targetNodeName, java.lang.String targetAgentName, java.lang.String sourceAgentName, java.lang.String topic, java.lang.String msg)                      This method is used to send message to a SAGE agent.</p> <p><b>Parameters:</b>                      targetNodeName - name of the node that hosts the agent to which the message needs to be routed                      targetAgentName - the agent to which the message needs to be routed                      sourceAgentName - the agent that sends the message                      topic - the topic of the message                      msg - message text</p>
boolean	<p><b>setUp</b>(nrl.sage.BehaviorInterface.Result result)                      This method is called when agent becomes active.</p> <p><b>Parameters:</b>                      result - SAGE result object to carry information about the result of method execution.</p>
boolean	<p><b>tearDown</b>(nrl.sage.BehaviorInterface.Result result)                      This method is called when agent becomes inactive.</p> <p><b>Parameters:</b>                      result - Java class that represents the outcome of the execution of the message method.</p>

**Table 4 Class QueryToInfer field summary**

<b>Field name</b>	<b>Field detail</b>
String[]	<p><b>qStrings</b>                      This is an array of query strings.</p>
Integer	<p><b>resultHashCounter</b>                      This is a counter that represents the number of SensorInfoUoC objects that are present in the resultHash HashMap</p>
HashMap< Integer, SensorInfoUoC>	<p><b>resultHash</b>                      This HashMap stores SensorInfoUoC type objects that are created as result of query to the sensor datastore.</p>

## 2.3 Class SensorDatastore

This class represents a SAGE behavior that ingests data from the sensor data set, stores it in appropriate data structures, and eventually responds to queries on the sensor data set (Tables 5 and 6). The sensor data set is in an Excel file where each sheet contains a combination/configuration of different sensor-type readings.

**Table 5 Class SensorDataStore method summary**

Modifier and type	Method and description
boolean	<p><b>action</b>(nrl.sage.BehaviorInterface.Result result)            This method is called intermittently by the agent after the agent becomes active.</p> <p><b>Parameters:</b>            result - Java class that represents the outcome of the execution of the action method.            result - SAGE result object to carry information about the result of method execution.</p>
void	<p><b>public void aggregateMap()</b>            Utility method that aggregates the various HashMaps into one HashMap. This message utilizes the result HashMap that is created from the query that is sent to the agent with SensorDataStore behavior.</p>
Double	<p><b>calculateUoI</b>(java.lang.Double colSensorVal, java.lang.Double maxColVal, java.lang.Double minColVal, java.lang.Double avgColVal, java.lang.Double stdColVal)            This method computes UoI information related to a sensor reading based on supplied parameters.</p> <p><b>Parameters:</b>            colSensorVal - sensor reading value            maxColVal - maximum value of the column containing the sensor type reading values            minColVal - minimum value of the column containing the sensor type reading values            avgColVal - average value of the column containing the sensor type reading values            stdColVal - standard variation value of the column containing the sensor type reading values</p> <p><b>Returns:</b>            Computed Uncertainty of Information (UoI) value.</p>
void	<p><b>checkforvalues</b>(java.lang.String config, java.lang.String tm, java.lang.String rg)            This method searches the sensor data store using the supplied parameters and creates resulting SensorInfoUoC objects.</p> <p><b>Parameters:</b>            config - Configuration of different sensor types            tm - Time stamp of sensor reading            rg - range of the time</p>

**Table 5 Class SensorDataStore method summary (continued)**

<b>Modifier and type</b>	<b>Method and description</b>
ArrayList<String>	<p><b>createArrayFromSensorObj(SensorInfoUoC se)</b> Utility method that takes SensorInfoUoC type object and creates an ArrayList of strings from its attributes.</p> <p><b>Parameters:</b> se - SensorInfoUoC Object</p> <p><b>Returns:</b> ArrayList derived from a "SensorInfoUoC" object</p>
ArrayList<Double>	<p><b>createArrayList_A_fromHashMap(java.util.HashMap&lt;java.lang.Integer,SensorInfoUoC&gt; m)</b> Utility method to extract readings from sensor type A, given a HashMap containing SensorInfoUoC objects</p> <p><b>Parameters:</b> m - HashMap containing SensorInfoUoC objects</p> <p><b>Returns:</b> ArrayList containing sensor readings</p>
void	<p><b>public void createDataStats()</b> This method is responsible for creating data statistics on different sensor type readings.</p>
Double[ ]	<p><b>findStats(java.util.ArrayList&lt;java.lang.Double&gt; ar)</b> This method computes different statistical values for the different sensor type readings.</p> <p><b>Parameters:</b> ar - ArrayList of sensor readings</p> <p><b>Returns:</b> Array of statistical values including maximum, minimum, average, and standard deviation values for sensor type readings.</p>
ArrayList<String>	<p><b>HashMapValuetoArrayListValue(java.util.HashMap&lt;java.lang.String,java.lang.Double[]&gt; m)</b> Utility method to create a string ArrayList from HashMap entries</p> <p><b>Parameters:</b> m - HashMap containing sensor information</p> <p><b>Returns:</b> String ArrayList of sensor information</p>
void	<p><b>iterateMap(java.util.HashMap&lt;java.lang.Integer,SensorInfoUoC&gt; mp)</b> Utility method to iterate HashMap containing SensorInfoUoC type objects</p> <p><b>Parameters:</b> mp - HashMap containing SensorInfoUoC type objects</p>
void	<p><b>loadhashmap()</b> This utility method parses through the Excel sheets and creates corresponding HashMap containing data from the sheets.</p>
boolean	<p><b>message(nrl.sage.BehaviorInterface.Message message, nrl.sage.BehaviorInterface.Result result)</b> This method is called upon a SAGE message arrival.</p>
HashMap<Integer, SensorInfoUoC>	<p><b>process_RegularSheet(org.apache.poi.ss.usermodel.Sheet st)</b> This method iterates through the rows of an Excel sheet.</p> <p><b>Parameters:</b> st - Excel sheet</p> <p><b>Returns:</b> HashMap containing "SensorInfoUoC" type objects</p>

**Table 5 Class SensorDataStore method summary (continued)**

<b>Modifier and type</b>	<b>Method and description</b>
boolean	<p><b>setUp</b>(nrl.sage.BehaviorInterface.Result result)            This method is called when agent becomes active.</p> <p><b>Parameters:</b>            result - SAGE result object to carry information about the result of method execution</p>
boolean	<p><b>tearDown</b>(nrl.sage.BehaviorInterface.Result result)            This method is called when agent becomes inactive.</p> <p><b>Parameters:</b>            result - SAGE result object to carry information about the result of method execution</p>
	<p><b>updateMap</b>(java.util.HashMap&lt;java.lang.Integer,<b>SensorInfoUoC</b>&gt; mp)            This method initiates computation of Uncertainty of Information (UoI) value based on sensor information.</p> <p><b>Parameters:</b>            mp - HashMap containing SensorInfoUoC type objects to which UoI information has to be added</p>

**Table 6 Class SensorDataStore field summary**

<b>Field name</b>	<b>Field detail</b>
HashMap< Integer, SensorInfoUoC >	<p><b>case1and2Hashmap</b>            This HashMap holds SensorInfoUoC type objects derived from one specific sheet.</p>
HashMap< String, Double[] >	<p><b>dataStats</b>            This HashMap contains metadata for columns of a sheet.</p>
HashMap< Integer, SensorInfoUoC >	<p><b>globalHashmap</b>            This HashMap holds all the SensorInfoUoC type objects derived from various sheets.</p>

### 3. Conclusion

This report describes an initial model on inferencing. The model will continue to be augmented with robust inferencing capabilities, so this document will need to be updated to reflect any new capability that gets added in the future.

## 4. References

---

1. US Naval Research Laboratory. SAGE user guide. Naval Research Laboratory (US); c2022 [accessed 2023 Aug 22]. <https://sageframework.com/>
2. Raglin A, Metu S, Lott D. Challenges of simulating uncertainty of information. International Conference on Human-Computer Interaction; Springer; 2020 July. p. 255–261.

## List of Symbols, Acronyms, and Abbreviations

---

SAGE      Sentry Agents

UoI      Uncertainty of Information

1 DEFENSE TECHNICAL  
(PDF) INFORMATION CTR  
DTIC OCA

1 DEVCOM ARL  
(PDF) FCDD RLB CI  
TECH LIB

2 DEVCOM ARL  
(PDF) FCDD RLA IC  
S METU  
A RAGLIN