NIKI: Network Issue Kinesthetic-Learning Information Tool

Author: Jordan Micah Bennett

**Contents**

1. NIKI’s diagnosis classes/types
2. How may you use NIKI?
   1. Peruse Mode (essential!)
   2. Faq Mode (optional!)
   3. Free Type Mode (optional!)
   4. Complain Mode (optional!)
3. NIKI Commands
4. Understanding NIKI’s knowledgebase. (technical topic)

**1. NIKI’s diagnosis classes**

NIKI architecture is “trained” to recognize the following class of network issues:

* General Security issues
* TCP/IP and other protocol issues
* Network performance issues
* Credential, permissions, and rights problems

**2.Using NIKI**

NIKI consists of 4 modes, the most important of which is **peruse mode**, where issues may be detected and solved. Issues are detected through user interaction, or rather when the user selects topics under any of NIKI’s five network issue classes in peruse mode.

**Faq mode** consists of descriptions of the classes within NIKI.

**Free-Type mode** shall grant the user the ability to state problems free hand, after which NIKI will attempt to return relevant results. This is similar to **peruse mode**, but less efficient.

**Complain mode** shall grant the user the ability to send problems to [nikitoolmail@gmail.com](mailto:nikitoolmail@gmail.com) automatically via NIKI.

**3.NIKI commands**

All queries are funneled to the user in the form of XPCE/UI layout, which grants the user the ability to solve issues in a window based graphical manner without running commands. **However**, all queries can be alternatively ran directly via SWI prolog IDE’s console, if the user prefers.

**This section will explain how to run commands to achieve equivalent UI experience, in a console manner.**

**COMMAND 0:**

all\_issues(ANY\_VARIABLE\_NAME).

**SAMPLE**:

all\_issues(A). (where A is a variable which stores result of query)

**DESCRIPTION**:

all\_issues returns a list of all of NIKI’s classes.

**COMMAND 1:**

troubleshoot\_question(SEARCH\_TITLE\_NAME,OUTPUT\_VARIABLE\_NAME):

**SAMPLE**:

troubleshoot\_question(general\_security,T).-->where general\_security is a fixed atom, and one of NIKI’s five network issue classes/diagnosis types, and T is a variable which collects results.

troubleshoot\_question returns a list of question markers or questions, based on input class type (question title).

**COMMAND 2:**

troubleshoot\_solution(target\_title,T).

**SAMPLE**:

troubleshoot\_solution(general\_security\_question\_marker\_0,X).-->where general\_security\_question\_marker\_0 is a fixed atom, and one of NIKI’s question markers which relates to a question titles,and T is a variable which collects the result of the query.

troubleshoot\_question returns a list of question markers or questions, based on input class type.

**4.Understanding NIKI’s Knowledgebase**

NIKI’s Knowledge base is beautiful and simple, and relies upon prolog’s logic mechanism to establish relationships based on **THREE** sets of facts/disjunctions which form the database.

NIKI is equipped with **THREE** powerful rules/commands/queries which may be used to make sense of or create relationships between the **THREE** sets facts mentioned above.

**The Three sets of facts exist in forms:**

**Fact zero form exists as collection of all available network issues diagnosable.**

**Here are 2 of 5 examples of fact zero’s form:**

niki\_issue\_title\_node(tcp\_ip\_protocol).

niki\_issue\_title\_node(network\_performance).

**Fact one form exists as collection of archive question markers vs issue types.**

**Here are 2 of 5 examples of fact one’s form:**

Example 1:

niki\_issue\_archive\_question\_node(general\_security\_question\_marker\_0,general\_security).

niki\_issue\_archive\_question\_node(general\_security\_question\_marker\_1,general\_security).

niki\_issue\_archive\_question\_node(general\_security\_question\_marker\_2,general\_security).

Example 2:

niki\_issue\_archive\_question\_node(tcp\_ip\_protocol\_question\_marker\_0,tcp\_ip\_protocol).

niki\_issue\_archive\_question\_node(tcp\_ip\_protocol\_question\_marker\_1,tcp\_ip\_protocol).

niki\_issue\_archive\_question\_node(tcp\_ip\_protocol\_question\_marker\_2,tcp\_ip\_protocol).

**Finally, the lowest in the hierarchy, fact two form exists as collection of archive answer markers vs question archive question answers.**

Here are 2 of 5 examples of fact two’s form:

Example 1:

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_0\_0,credential\_permission\_rights\_question\_marker\_0).

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_1\_0,credential\_permission\_rights\_question\_marker\_0).

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_2\_0,credential\_permission\_rights\_question\_marker\_0).

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_0\_1,credential\_permission\_rights\_question\_marker\_1).

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_1\_1,credential\_permission\_rights\_question\_marker\_1).

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_2\_1,credential\_permission\_rights\_question\_marker\_1).

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_0\_2,credential\_permission\_rights\_question\_marker\_2).

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_1\_2,credential\_permission\_rights\_question\_marker\_2).

niki\_issue\_archive\_answer\_node(credential\_permission\_rights\_solution\_text\_2\_2,credential\_permission\_rights\_question\_marker\_2).

**Example 2:**

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_0\_0,initial\_configuration\_question\_marker\_0).

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_1\_0,initial\_configuration\_question\_marker\_0).

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_2\_0,initial\_configuration\_question\_marker\_0).

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_0\_1,initial\_configuration\_question\_marker\_1).

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_1\_1,initial\_configuration\_question\_marker\_1).

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_2\_1,initial\_configuration\_question\_marker\_1).

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_0\_2,initial\_configuration\_question\_marker\_2).

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_1\_2,initial\_configuration\_question\_marker\_2).

niki\_issue\_archive\_answer\_node(initial\_configuration\_solution\_text\_2\_2,initial\_configuration\_question\_marker\_2).

**The Three rules exist in forms:**

The troubleshooting question function returns NIKI

question, based on title input knowledgebase title/diagnosable class.

**Rule form zero:**

**COMMAND :**

troubleshoot\_question(target\_title,GENERIC\_VARIABLE).

**SAMPLE :**

troubleshoot\_question(general\_security,X).

**STRUCTURE:**

troubleshoot\_question(SEARCH\_TITLE\_NAME,OUTPUT\_VARIABLE\_NAME):-

SEARCH\_TITLE\_NAME = IssueTitleItem,

setof(IssueArchiveQuestionItem,niki\_issue\_archive\_question\_node(IssueArchiveQuestionItem,IssueTitleItem),OUTPUT\_VARIABLE\_NAME).

**Rule form one:**

The troubleshooting solution function returns any NIKI

relevant answer records based on input question marker. Question markers are just indexed questions in the database

**COMMAND :**

troubleshoot\_solution(target\_title,GENERIC\_VARIABLE).

**SAMPLE :**

troubleshoot\_solution(general\_security\_question\_marker\_0,X).

**STRUCTURE:**

troubleshoot\_solution(SEARCH\_QUESTION\_NAME,ANY\_VARIABLE\_NAME):-

SEARCH\_QUESTION\_NAME = IssueArchiveQuestionItem,

setof(IssueArchiveAnswerItem,niki\_issue\_archive\_answer\_node(IssueArchiveAnswerItem,IssueArchiveQuestionItem),ANY\_VARIABLE\_NAME).

**Rule form two:**

Reveals all network issue types.

**COMMAND:**

all\_issues(GENERIC\_VARIABLE).

**SAMPLE:**

all\_issues(X).

**STRUCTURE:**

all\_issues(ANY\_VARIABLE\_NAME):-

setof(IssueTitleItem,niki\_issue\_title\_node(IssueTitleItem),ANY\_VARIABLE\_NAME).