# 1 start BeehiveZ

## 1.1 on windows platform

execute "run.bat"

## 1.2 on linux platfrom

execute "run.sh"

before that, you must be sure that Graphviz has been installed, and chmod the "run.sh" to be executable

# 2 Functions

E:\UDMS\代码相关\BeehiveZ2\src\resources\images\Icon_ModelIO.gifmodel management

E:\UDMS\代码相关\BeehiveZ2\src\resources\images\Icon_MiningEvaluate.gifprocess mining algorithm evaluation

E:\UDMS\代码相关\BeehiveZ2\src\resources\images\Icon_SimilarityMetric.gifprocess similarity metric

E:\UDMS\代码相关\BeehiveZ2\src\resources\images\Icon_PerformanceWatch.gifperformance watch

E:\UDMS\代码相关\BeehiveZ2\src\resources\images\Icon_IndexManagement.gifindex management

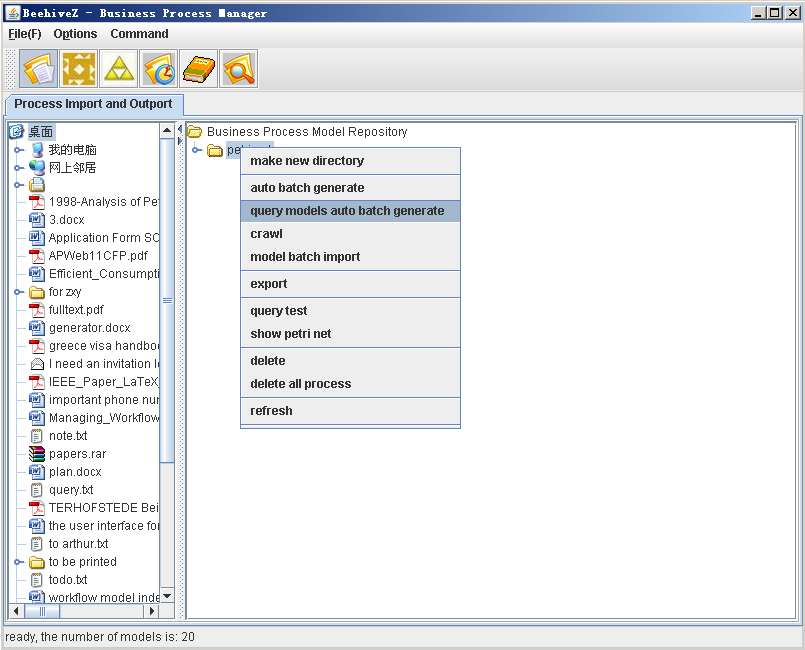
E:\UDMS\代码相关\BeehiveZ2\src\resources\images\Icon_ModelQuery.gifmodel query

Note: At the moment, all the models in this system must be Petri net, which is stored as PNML

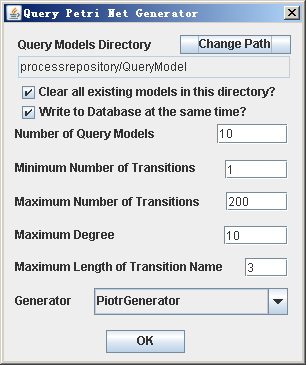
## 2.1 model generation

### 2.1.1 query model generation

Right click, you will see as follow



Choose “query models auto batch generate”, you will see as follow



“query models directory”: the directory to store the models generated

“clear all existing models in this directory”: determine whether the models already existing in the directory would be deleted first.

“write to database at the same time”: determine whether the models generated would be added into the database at the same time

“Number of Query Models”: the number of models will be generated.

“Minimum Number of Transitions”: in one model, the minimum number of transitions it can contain

“Maximum Number of Transitions”: in one model, the maximum number of transitions it can contain

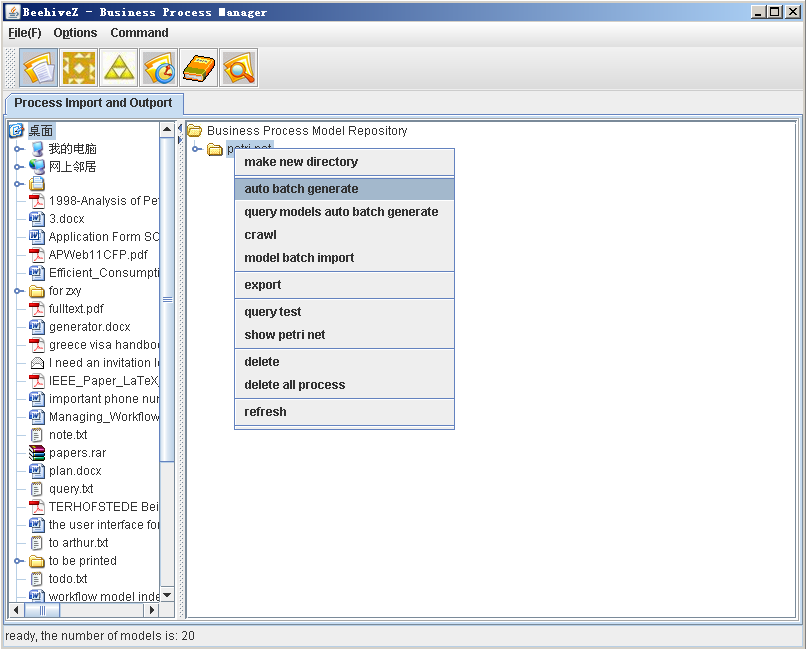
“Maximum Degree”: the maximum input/output degree, only works for GWFGenerator

“Maximum Length of Transition Name”: every transition have a string label, the maximum length can be determined by this parameter

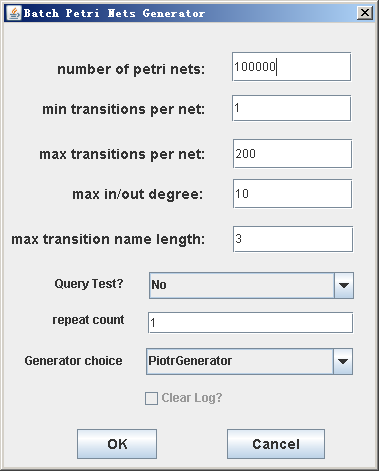
“Generator”: there are 4 generators can be used.

1. GWFGenerator: can generate sound structured workflow-nets;
2. PiotrGenerator: can generate sound workflow-nets with communication and synchronization;
3. MurataGenerator: can generate sound workflow-nets with Murata’s rules;
4. MoeYAWLGenerator: can generate sound YAWL models with Moe’s rules;

### 2.1.2 large number of models generation



Choose “auto batch generate”, you will see as follows



“number of Petri nets”: how many models will be generated

“min transitions per net”: the minimum number of transitions every generated model can contain

“max transitions per net”: the maximum number of transitions every generated model can contain

“max in/out degree”: the max input/output degree for every node in the generated models

“max transition name length”: every transition has a string label, the max length of these strings

“Query Test”: during the model generation, can some queries be conducted? (1)”No”, no queries will be conducted during the models generation; (2)”query using specific models”, user can configure a directory where all the models can be used as queries;(3)”query using the model added”, every model generated can be used as a query after it is added into the repository

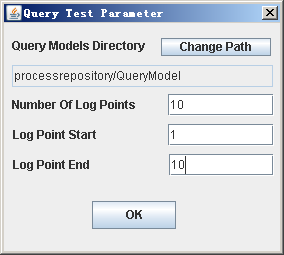
“repeat count”: how many times a query can be conducted, only works for “query using specific models”. It can be used to compute the average time cost.

“Generator choice”: there are 4 generators can be used.

1. GWFGenerator: can generate sound structured workflow-nets;
2. PiotrGenerator: can generate sound workflow-nets with communication and synchronization;
3. MurataGenerator: can generate sound workflow-nets with Murata’s rules;
4. MoeYAWLGenerator: can generate sound YAWL models with Moe’s rules;

“Clear Log”: when queries are conducted during the models generation, some data such as time consumed will be logged, this checkbox can be used to clear the logs generated before

If we use specific models as query during the models generation, more parameters should be configured as follows.



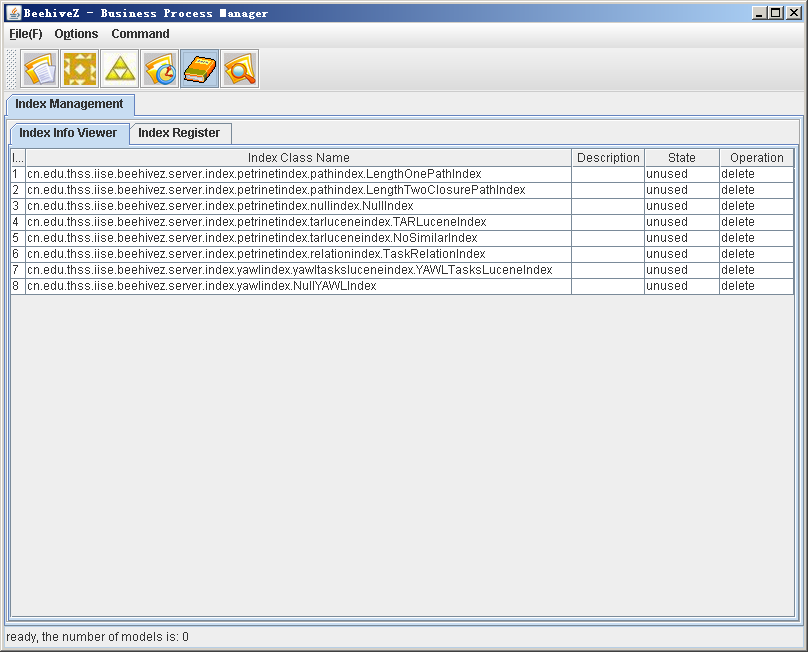
“Query Models Directory”: the directory where all models can be used as query

“Number Of Log Points”: how many points will be logged during the models generation with queries conducted.

“Log Point Start”: the first point to log

“Log Point End”: the last point to log

## 2.2 index management

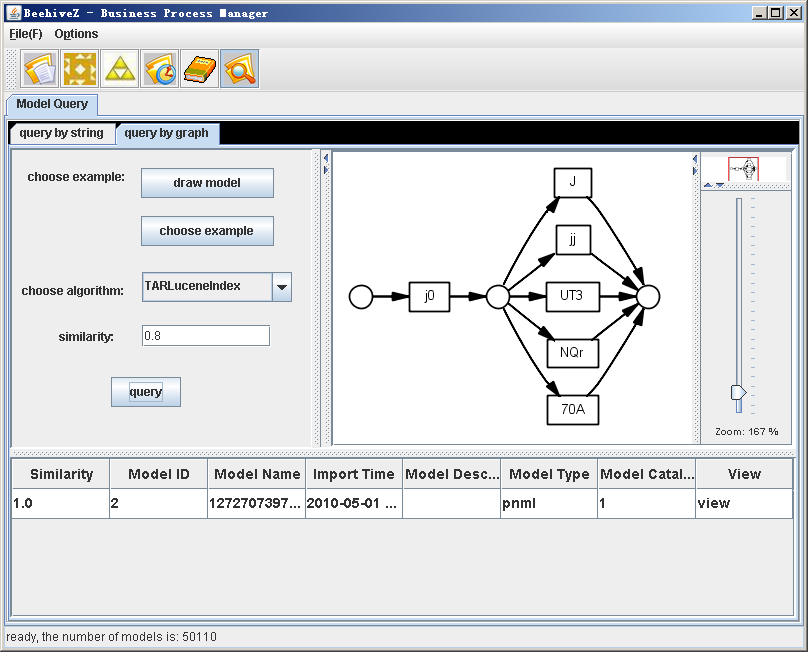


You can click the cell of state to use some index or unuse some index.

The existing indexes are as follows.

1. Petri net index
2. LengthOnePathIndex: for exact Petri net retrieval, only index on length one pathes
3. LengthTwoClosurePathIndex: for exact Petri net retrieval, index on length one and two pathes
4. NullIndex: for exact Petri net retrieval, no index for sub-graph query
5. TARLuceneIndex: for similar Petri net retrieval, index on the tars
6. NoSimilarIndex: for similar Petri net retrieval, no index for similar query based on tars
7. TaskRelationIndex: task relations index, can be used to query Petri nets based on ordering relations
8. YAWL index
9. YAWLTasksLuceneIndex: YAWL tasks index, for exact YAWL model query
10. NullYAWLIndex: for exact YAWL model query, no index for sub-graph query

## 2.3 query by example



“draw model”: use PIPE to draw a model.

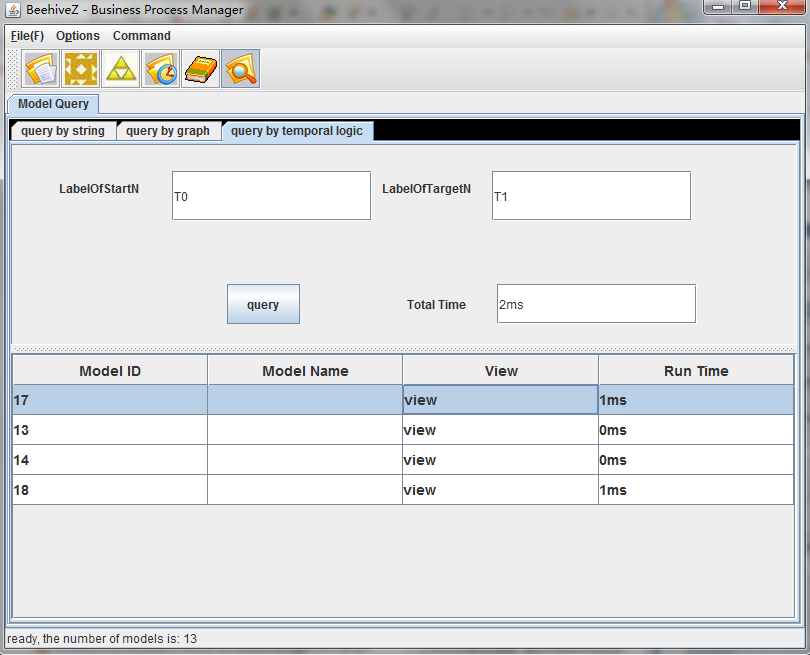
“choose example”: select a model from disk and use it as a query.

“choose algorithm”: choose an index algorithm, all the algorithms here are usable.

“similarity”: set the similarity for similar retrieval. For exact retrieval, the similarity is 1

When you click the button “query”, you can get the result (descending order according to similarity) in the below list, click the cell of “View”, you can view the result model.

## 2.4 query by temporal logic



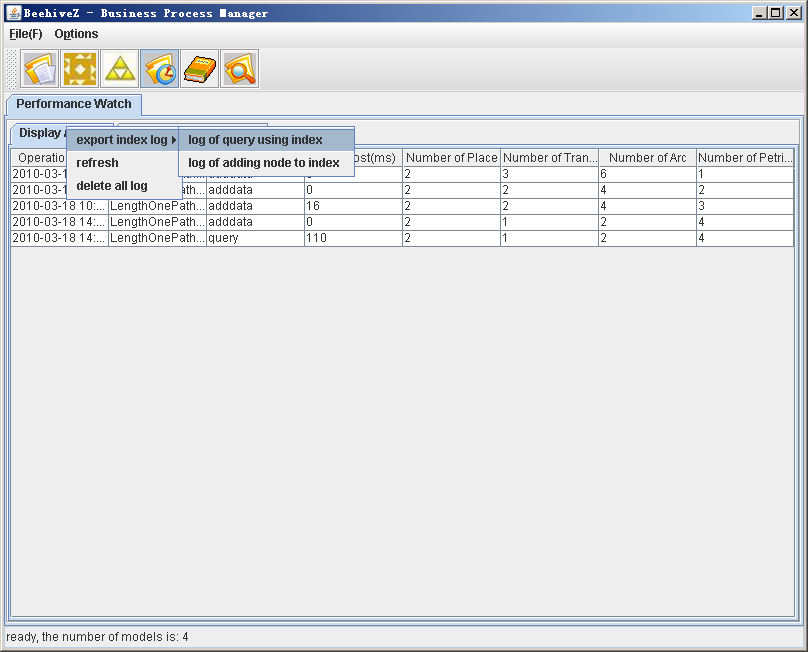
“LabelOfStartN”: the label of the start node.

“LabelOfTargetN”: the label of the target node.

“TotalTime”: the time of query.

When you click the button “query”, you can get the result in the below list, click the cell of “View”, you can view the result model.

## 2.5 export log

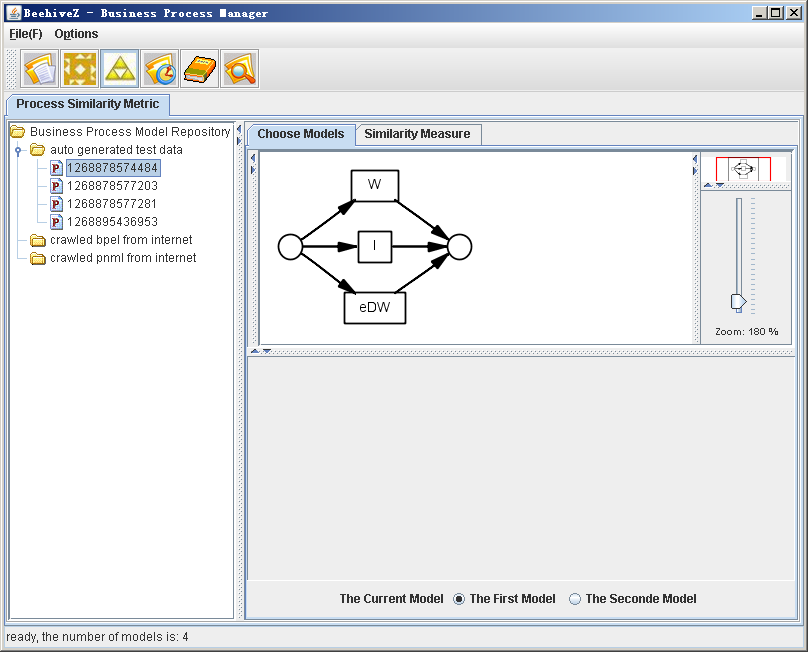


You can export the log from database, which log the time consumed for adding a new model into the repository and the time consumed for query.

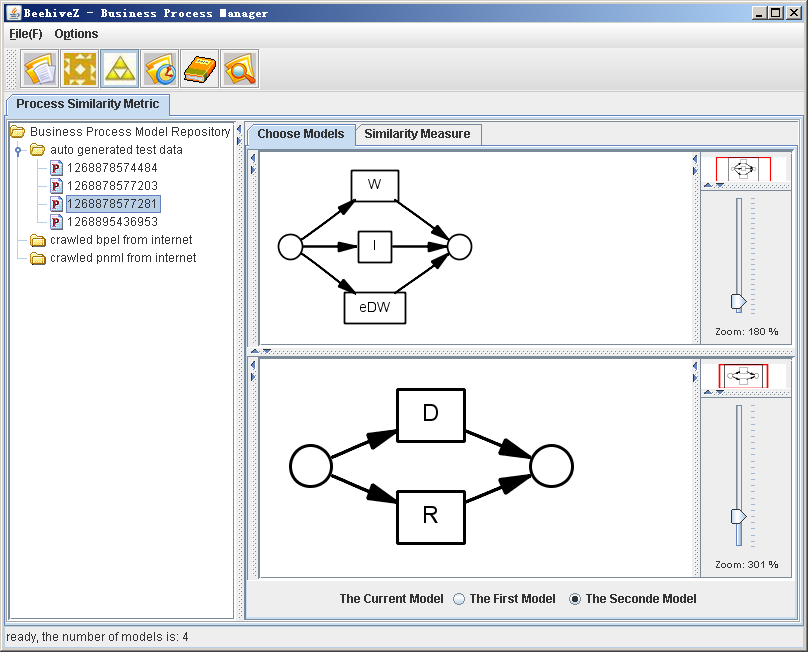
For other logs, such as search time consumed, candidate size, storage size, you can find them in the directory “processrepository\index\indexlog”

## 2.6 process similarity measure

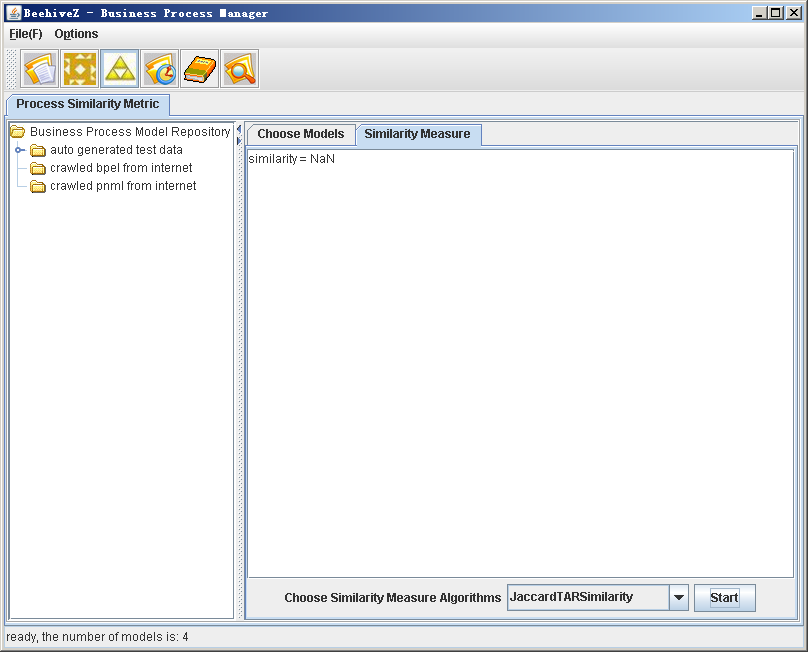
Choose the first model as follow.



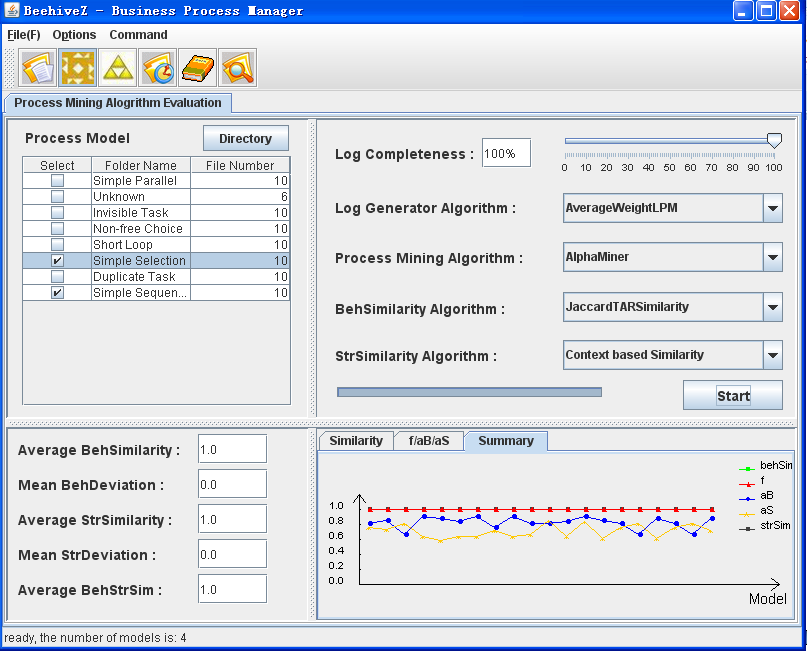
Choose the second model as follow



Measure the similarity as follow. You can choose different similarity measure algorithms.



## 2.7 process mining algorithm evaluation



“Directory”: configure a directory containing some folders organized according to their typical

control-flow constructs, e.g., short-loop, non-free choice.

“Log Completeness”:choose the completeness of logs based TAR.

“Log Generator Algorithm”: choose one log generator algorithm.

“process Mining Algorithm”: choose one process mining algorithm.

“BehSimilarity Algorithm”: choose one BehSimilarity algorithm.

“StrSimilarity Algorithm”: choose one StrSimilarity algorithm.

Click the button “Start”, you can see the result in the below window, including:

1.average BehSimilarty,mean BehDeviation,average StrSimilarity,mean StrDeviation,

2.Similarity, f/aB/aS, and summary,

you can also find the result in the file “result.xls”.

## 2.8 Experiments for <TAR\*: an Improved Process Similarity Measure Based on Unfolding of Petri nets>

You can find the two experiments for paper <TAR\*: an Improved Process Similarity Measure Based on Unfolding of Petri nets> in the following classes respectively:

src\cn\edu\thss\iise\beehivez\server\basicprocess\occurrencenet\experiment\_2.java and

src\cn\edu\thss\iise\beehivez\client\ui\similaritymetric\experiment\_3.java

# 3 System config

## 3.1 database config

Using “db.conf”

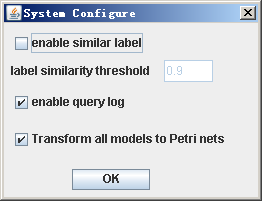
Now, Derby, PostgreSQL and MySQL are supported. You can refer to “derby.conf”, “postgresql.conf” and “mysql.conf”.

The sql scripts of create databases can be found in the directory of sql.

## 3.2 label similarity config

Using “system.conf”

You can change the config using menu <command>/<System Configuration> as follows.



**Note: If the “Transform all models to Petri nets” is unchecked, when models are added to the repository, the Petri net indexes will not be updated.**

## 3.3 path-index config

Using “PathIndexConfig.ini”

# 4 Query language for TaskRelationIndex

* <Expression> ::= <AndExpression> (<Or> <AndExpression>)\*
* <AndExpression> ::= (<RelationExpression> | <ExistExpression> | "("<Expression>")") (<And> (<NotExpression> | <RelationExpression> | <ExistExpression> | "("<Expression>")"))\*
* <NotExpression> ::= <Not> (<RelationExpression> | <ExistExpression> | "("<Expression>")")
* <ExistExpression> ::= <Exist>(<Activity> | "("(<Activity>)+")")
* <RelationExpression> ::= <Activity> <Relation> <Activity>
* <Exist> ::= "exist"
* <Activity> ::= "\"" (~["\""])+ "\""
* <Relation> ::= (<ParallelWith> | <Exclude> | <Precede>)
* <ParallelWith> ::= "parallel with" | "=="
* <Exclude> ::= "exclude" | "##"
* <Precede> ::= "precede" | "->"
* <And> ::= "and" | "&&"
* <Or> ::= "or" | "||"
* <Not> ::= "not" | "!"

# 5 Others

## 5.1 metrics

You can use the menu <command>/<Model metrics> to collect the metrics data from Petri nets or YAWL models in database or file system as follows.

