

CLAIMS

What is claimed is:

1. A system for configuring and executing a workflow based on micro-services, comprising:
at least one Graphical User Interface (GUI) for displaying one or more nodes;
at least one storage medium for storing at least one workflow process configured using the GUI along with a unique workflow identity; and
at least one execution unit for retrieving the workflow process based on the unique workflow identity from storage medium and processing the workflow to produce a result;
wherein, the GUI is configured to allow a user to drag and drop the nodes and link the nodes to configure the workflow process;
wherein each node in the workflow process waits in a queue for execution in the execution engine and said queue operable by an asynchronous message mechanism.
2. The system of claim 1, wherein the nodes are connected in a desired relationship to form the workflow process.
3. The system of claim 2, wherein the relationship between two nodes is stored as a from-to pair.
4. The system of claim 1, wherein each node comprises a unique process node identity (ID).
5. The system of claim 4, comprising one or more of:
a basic node code and a current timestamp; and
a workflow ID and a workflow node ID.

6. The system of claim 1, wherein each node is designed to have a different status change and a decision change.
7. The system of claim 1, wherein each node configured in the workflow process has a corresponding node executor.
8. The system of claim 1, wherein an execution weight of each node in the execution engine is equal.
9. The system of claim 1, wherein each node is queued up again after completion of execution.
10. The system of claim 1, wherein the execution unit applies a validation to the workflow process based on a pre-defined validation rule.
11. The system of claim 1, wherein the execution unit receives an assignment, and processes the assignment to produce a result.
12. A method of configuring and executing a workflow based on micro-services, comprising:
 - displaying, on at least one Graphical User Interface (GUI), one or more nodes and allowing a user to drag and drop the nodes and link the nodes to configure at least one workflow process;
 - storing, in at least one storage medium, the workflow process with a unique workflow identity;
 - retrieving, by at least one execution unit, the workflow process based on the unique workflow identity from the storage medium; and
 - processing, by the execution unit, the workflow process to produce a result;wherein each node in the workflow process waits in a queue for execution in the execution engine and said queue operable by an asynchronous message mechanism.

13. The method of claim 12, comprises:

receiving, by the execution unit, an assignment; and
processing, by the execution unit, the assignment to produce a result.

14. The method of claim 12, further comprising:

allowing the user to return to any executed returnable node;
checking, by the execution engine, for a return-able node by:
finding, previous executed nodes of a current node using node relations in process
nodes details and executed node details;
sorting, previous executed nodes based on execution time and finding the latest
executed node;
applying, validation for the latest executed node based on a pre-defined validation
rule;
marking, the latest executed node as return-able, if it passes the validation;
sorting, the previous executed nodes based on execution time and finding a latest
executed node;
applying, validation for the latest executed node based on the pre-defined
validation
rule; and
marking, the latest executed node as return-able, if it passes the validation.

15. The method of claim 12, wherein each node configured in the workflow process
has a corresponding node executor and the node executor is developed by:

defining, an abstract class implementing an execute method;
defining, the node executor for each type of node by extending the abstract class;
and defining, an execute factory with a register to put all the node code and
executor pairs in a map,

wherein the execute method of the abstract class comprises checking a status and initializing work, executing work, and changing the status and handling an error.

16. The method of claim 12, wherein each node is designed to have a different status change and a decision change during execution of the workflow.

17. The method of claim 12, further comprising:

assigning an equal weight to each node and execution of each node in the order of sequence entered in an execution queue.

18. The method of claim 12, further comprising:

queuing up the nodes again after completing execution of the node;

removing automatically, by the execution unit, the node from the execution queue;
and

moving the node to a task pool, when the execution of the corresponding node is blocked or when permission of corresponding node is revoked.

19. A computer readable medium storing a program comprising instructions that, when executed by a system, cause the system to perform the method for configuring and executing a workflow based on micro-services of any of one claims 12 to 18.