Taskoids: A Formal Definition

Dr Bheemaiah, Anil Kumar , Seattle, W.A 98125 USA.
miyawaki@yopmail.com

Abstract:
A formal definition of taskoids with a future market in trading in work credits with carbon credits and the happiness index clock. Taskoids are defined with a machine genome basis as the quantification of the automation of tasks. They stem from the natural programming, mathematical programming and automated persistence cloud model of computing with computer-assisted code generation. In this paper, we formally define a taskoid and use AWS infrastructure as a service to define IAC rules, encrypted in machine genome to customize solutions using AWS for tasks using a machine genome to transcript YAML or JSON representations.

Keywords: Taskoids, Quantification, machine genome, cloud computing, YAML, JSON

What:

- Formal Definition of Taskoids.
- The formal definition of machine genome.
- Definition and classification of task computability as the formulation of automation.
- Example of an AWS CloudWatch and CloudFormation based JSON based Taskoid.

How:
Taskoids are meta-programs that use existing codebases and automated coding to encrypt in machine genome a quantIfication of automation.

Why:
Similar to Soul Machines and Digital DNA of Digital Human designs, we create taskoids with machine genome to easily or automatically configure IAC to an application for a class of task computability.

Summary:

Main Points:
Two sets of theorems are presented:

1. Existence theorems.
2. Classification theorems.

The classification ZT and QT are defined for the expert system of quantum and stochastic algorithms for many taskoids, for which an expert system is described.

Applications:
RPA and DPA based automation and work automation using Alexa Business skills for all tasks as proven by the existence theorem for SaaS-based solutions to automation., hence taskoid futures can be written for all possible applications needing work automation. The classes ZT and ST, define new algorithms with an expert system, which is searchable for possible combinations of taskoids for work automation.

Code Base:
GitHub Repository
Website

Introduction.

Machine evolution is distinct from natural evolution, in intentional evolution, From form defined by function, and ontologies of space-time and events, and extended to multifunctional architectures, with a prime directive of machine evolution.

Anthropological research usually credits tool usage as intelligence, true intelligence in machine evolution of singularity is defined by the Turing test, we define singularity#1 and transcription of machine genome as algorithms for machine evolution as formulated in the algorithms of evolutionary computing.

The proof of singularity#1, defined as the design of machines, circuits, algorithms, and code, by algorithms is by evolutionary computing. As opposed to algorithms like genetic algorithms, the intention in machine evolution is seen as adaptability over a natural evolution.

This is the basis of work automation using a SaaS-based infrastructure and algorithms.

Work and tasks are quantified as taskoids, encrypted as machine genome, transcribed as intention, to a use case, proving singularity#1.

Early machine evolution algorithms, were based on genetic algorithms, with a parameter-based machine genome, a transcription model and a fitness algorithm. with mutation and crossover defining the evolution, this is now replaced by intentional evolution, an innovation pipeline of taskoids, a diversity of them with an MFA II framework, based on a SaaS framework.

Problem Definition.

A Taskoid is defined as the Quantification of work automation in a SaaS-based framework. There exist Existence theorems. and Classification theorems. For flex-rules and algorithms in classes ZT and QT with an expert system for the usage of these algorithms, these are classification ontologies and existence proofs of polynomial-time algorithms, the proof is by a framework of transcription of a machine genome framework.

Transcription Theorems:
There always exists a transformation T -> (R,S),(Rep)
Where Rep is a representation like JSON, YAML or XML and TS is the intentional taskoid in CI/CD on the cloud infrastructure.
R is sequence transcription.

Background.

SaaS-based automation and low code solutions were invented in the previous decade and are now by solutions like AWS Blueprints, Pega Automation, and Twilio communications to name a few. (“Digital Process Automation” 2018, “Gartner Magic Quadrant for Low-Code Application Platforms 2019 | Appian Google Ad” n.d.) Rule-based automation systems have been enhanced with rule extraction and rule revision using deep learning networks. Much work has been done on measuring the automation efficiency of RPA systems and the modification of rules in less performing systems,(Ishikawa, n.d.) we define a new concept of RPA and DPA systems with flexible rules called flex-rules in a companion publication.

<original-contribution>
Formal Definitions:
Axioms:
{
A Taskoid T is defined as a quantification of automation encryptable in a finite alphabet $A0$, $A1$, $...An$, forming the machine genome.

A Sequence S = [s0, s1,.... sk] forms the atomicity of the machine genome.

There exists a transformation from a sequence to an object representation like YAML or JSON or a tag representation like XML for the transcription of the sequence.

There exists an object representation of the requirements for a problem that integrates into sequence transcription, R.

Taskoids are formalized by task computability in P, QP, and NP, implying polynomial-time solutions for automation or for NP/QP completeness, an approximation.

Automated persistence is always implemented in cloud-based automation.

A taskoid belongs to an innovation space II, and has a lifetime and a half-life defined.

The efficiency of a taskoid can be measured by the innovation diffusion using percolation theory.

An innovation pipeline is in the evaluation of competing Taskoids and the determination of the need for upgradation with a new family of Taskoids.

}

Theorems =
{
Transcription Theorems:

1. There always exists a transformation T -> (R,S),(Rep)
   Where Rep is a representation like JSON, YAML or XML.

2. Classification: A taskoid belongs to the TC class QT if there exists a quantum algorithm like a Q experience as one or more atomic cloud functions, which can be linked. Similarly, ZT if there exist stochastic cloud functions, with the existence of true random numbers in native coding in the cloud.

}

Example from AWS CloudWatch and CloudFormation.

We delineate an example from a medium tutorial.(GarryPas 2019). The eaxmple is IAC for a Taskoid with the task: A simple Lambda function in Amazon's AWS, that runs on a schedule, pulls some data down from an external API, and stores it in an S3 bucket. CloudFormation template looks like this:

```
{
 "AWSTemplateFormatVersion":
"2010-09-09",
 "Resources": {
  "PullMarkets": {
   "Type": "AWS::Lambda::Function",
   "Properties": {
    "FunctionName": "PullMarkets",
    "Handler": "lambda/index.handler",
    "Role": {
     "Fn::GetAtt": [
      "PullMarketsRole",
      "Arn"
     ]
    },
    "Code": {
     "S3Bucket": "my-lambda-functions",
      "S3Key": "PullMarketCode"
    },
    "Runtime": "nodejs8.10",
    "Timeout": 300
   }
  }
 }
}
```

The following template is used to create a machine genome format in JSON, which has the parameters, [AWS Lambda, Trigger in CloudWatch(with event rules), IAM authentication]
This forms the genome, [template, parameters]
With a transcription mechanism of a Map M defined to take the parameters and preprocess the template to the resulting CloudFormation JSON objects for CI/CD. See (GarryPas 2019)for a complete description of the intermediate steps. We leave the delineation of M as an exercise for the reader.

Expert System Flex-Rules.

A Website on Github.io for the contributions by the AWS community to Stochastic and Quantum algorithms for taskoid creation and the flex-rules needed for creating taskoids with them.

Story:

UOL is a graduate EdTech program with 600 and 700 level courses, while the 700 level courses involve scientific publications and reports, the 600 level courses are based on the Udacity Learning Model with live projects. They involve the authoring of Taskoids to be available through marketplaces like AWS or other clouds as futures.

</original-contribution>

Discussion.

The existence theorem proves the existence of an exact or an approximation algorithm for a taskoid for every work automation task, proving the universality of flex automation, which is DPA. The efficacy and need for the quantum cloud and/or stochastic algorithms with true random number generators are added through a website based repository. The next paper illustrates with examples, the machine genome algorithms and the algorithms for customization of the taskoids to create YAML and JSON description files for IAC taskoids.

Future Work.

Future work would involve creating a conversational UI for the expert system of ZT and QT flex-rules.

References.

"Digital Process Automation." 2018. February 13, 2018. https://www.pega.com/products/pega-platform/digital-process-automation.

GarryPas. 2019. "Create and Deploy an AWS Lambda Function with AWS CloudFormation." Medium. Medium. March 22, 2019. https://medium.com/@garry.passarella/create-and-deploy-an-aws-lambda-function-with-aws-cloudformation-583d5a2b1df0.

"Gartner Magic Quadrant for Low-Code Application Platforms 2019 | Appian Google Ad." n.d. Appian. Accessed August 25, 2019. https://www.appian.com/resources/gartner-magic-quadrant-for-enterprise-low-code-application-platforms-2019-google/.

Ishikawa, M. n.d. "Neural Networks Approach to Rule Extraction." *Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*. https://doi.org/10.1109/annes.1995.499427.