RUDIMENTARY SOLUTION FOR REFLEX ARTIFICIAL INTELLIGENCE IN DISTRIBUTED COMPUTING

Gandhi Sivakumar¹, G. Arumugam²

¹Research Scholar, Department of Computer Science, School of IT, Madurai Kamaraj University, Madurai, Tamil Nadu, India.

²Senior Professor and Head (Retd), Department of Computer Science, Madurai Kamaraj University, Madurai, Tamil Nadu, India.

Email: ¹gandhisiva@gmail.com, ²gurusamyarumugam@gmail.com

Corresponding Author: Gandhi Sivakumar

https://doi.org/10.26782/jmcms.2020.04.00017

Abstract

Artificial Intelligence (AI) technology has been adopted rapidly in the industry. Various research initiatives have been carried out to innate the AI system characteristics as humans. In our concept paper [VI] we disclosed the “Reflex layer” to mimic human systems. A reflex layer would have the ability to differentiate the repetitive stimuli, its related responses and ability to process this through a separate layer.

We discussed the key characteristics of reflex features of the following AI capabilities:

- The vision interface
- The audio interface
- The kinematic interface
- The sheath interface
- The core layer

In this paper we baseline the scope to core and kinematic interface; elaborate key characteristics, provide solutions and results.

Keywords: Artificial Intelligence, Distributed Artificial Intelligence, Reflex AI.

I. Introduction

AI systems span a spectrum of interfaces and enabling capabilities. Interfaces encompass entities from primitive text based chat bots, Virtual to dimensional interactive Avatars to Kinematic Robots with variegated organs and associated
actions. Interfaces portray the external manifestation and intelligence of the enabling capabilities at the back end.

Enabling capabilities are built on an ecosystem; data and algorithms operating on the data serve as fundamental building blocks in the ecosystem. Other key capabilities include the necessary resources (Compute, RAM, Physical layer) for execution.

AI solutions can be defined as an agglomeration of Interfaces and Algorithms acting on diverse data as specified below:

- Regular data sets
- Stray data i.e. unstructured data

Algorithms could be supervised (which require pre-training) or unsupervised (which do not require training). Algorithms may be visualized as rudimentary entities (RE) in the AI agglomeration.

We discussed in the concept paper [VI] that AI systems are built inline to mimic humans or tend to acquire characteristics of humans. We also discussed about Reflex action layer in AI. The scope of this paper being the “Solution for Reflex actions”, in section II we set the context by elaborating the key types of reflex actions of humans. In section III we categorize the RE entities, analyze the parameters of reflex actions. In section IV we publish the “heuristics” and finally conclude.

II. The Human Reflex Stack – conceptual view

Figure 1 shows the logical view of the “Human Reflex ”.

The human nervous system is able to perform two types of actions:

- Voluntary Actions – Actions which are done by the brain consciously by taking time to interpret the stimulus; such actions vary depending upon other priorities.
Involuntary or reflex Actions – Actions which are done by the nervous system unconsciously with no thinking thus faster; such actions are always the same. Reflex actions span a dichotomy;
- The ones controlled by the brain are termed as “cerebral reflex actions”;
- The ones controlled by the spinal cord are termed as “spinal reflex actions”.

The Reflex Arc of the Human Nervous system- Recap

In humans the reflex arc controls the reflex as the sensory neurons pass through the spinal cord. Spinal cord takes decisions for emergencies and also for repeated stimuli. (Note: Brain as well receives sensory signals and may take further conscious actions). This behavior ensures the following benefits:

- Faster responses to emergency situations
- Reduce load to the brain for responses to repeated stimuli
- Respond to stimuli when the brain is dormant (i.e when at sleep).

III. The AI Reflex Solution

We term the solution for the earlier disclosed concept as “The AI Reflex Conglomerate (AI_RC)”. AI_RC mimics the human reflex architecture. AI_RC performs actions both for anomalies and habitual scenarios. (Note: Anomaly means unusual scenarios and habitual means repeated scenarios). Figure below shows the AI_RC context.
AI_RC key components:

AI SPINE:  
AI Spine is composed of a constellation of AI Reflex Arcs. Each AI Reflex Arc consists of Input/Output interfaces and Arc Business processes.

Input Interface: Input interface accepts the features collected from external sources. Equating this to the human system this resembles the sensory nerves which collect stimuli. The features in parallel get sent to the AI Brain (Note- We detail the AI brain is detailed later in this subsection).

Arc Business Processes (ABP):  
ABP encompasses multiple processing units, decision points start and end units.

Threshold Identification Unit (TIU) :  
TIU is the first processing unit of the ABP. Trained models for “anomaly” detection reside in this unit and identify abnormalities. In humans instinct intelligence exists by birth and as the human grows this intelligence gets built through various factors including intelligence acquired through experience by exploration, involuntary effects or knowledge acquisition from peer humans or other.

Figure 3: AI Reflex Conglomerate and Human Reflex context – Zoomed in View

Copyright reserved © J. Mech. Cont. & Math. Sci.
Gandhi Sivakumar et al
We specify the following skill acquisition methods for our solution:

Instinct intelligence which gets developed based on the product specifications for a product in scope, for example an industry pump. As the pump is put to work, intelligence can be acquired like humans based on the effect for example with the snap shot of features when the pump was heated up and went faulty. On the other hand the TIU can be trained by a Data Scientist traditionally as well. The scope of skill acquisition is beyond the scope of this paper however.

In normal situations no action is taken. In case of anomaly the signal gets passed to the Arc- action foot print Unit.

**Arc Action foot print Unit for Anomaly (AAFPU_A):**

AAFPU_A isthe most important unit of the AI Reflex Arc. This unit possesses the intelligence of *Arc action foot print*. We define the Arc action foot print as the domain of control by the Reflex Arc i.e the actions which can be taken by the Reflex Arc. Considering an example from human system, immediate withdrawal of hand when a hot spot is sensed is within the foot print of Spine Arc. Other actions for example triggering the vision to focus on the hot spot or other actions is beyond its foot print.

Similarly AAFPU_A’s action foot print is confined within its boundaries of limitation. For example shutting down the pump could be within its limits but reducing the speed or closing an inlet valve might be actions beyond its portfolio.

**Output Interface:**

Output interfaces pass on the trigger actions to the Arc action which acts on the decisions.

**Environment:**

The characteristics of the AI Reflex Arc is to process input features with high speed and act fast for Anomaly. We propose the AI Reflex Arc anomaly path is handled by high processing compute with RAM capacity and physical layer bandwidth. Additionally we propose that the AI Reflex needs to be layered closer to environment, it needs to act on.

*Figure 4: AI Arc and Environment context*
Considering our pump example earlier, AI Arc should be layered as closer as it could be to the pump for faster decisions in case of anomaly. For example AI Arc should be layered as an embed within the pump or the cloud infrastructure whichever is the closest one.

Arc- Action footprint for Habitual process (AAFPU_H):

AAFPU_H is another important unit of the Reflex Arc. One can view the benefit of this unit as not only to reduce the load on the AI Brain but also to function even when there is no connectivity to it. The AAFPU_H possesses intelligence to recognize repetitive feature sets and value ranges mapped to the same or slightly varying but similar actions. In humans this is almost empty at the time of birth but gets built in due course even at infant stages. A classic example would be the baby responding to repeated sounds. In our solution this intelligence gets built in through Habitual Identification Unit of AI Brain detailed in the subsequent sections.

Environment:

Unlike Anomaly footprint’s environment, AAFPU_H can be hosted in a normal environment with normal Compute/ RAM and Physical layer characteristics.

The AI Brain:

AI brain is the core central system and possesses the conscious capability. It normally is layered far to the AI Spine and possesses the following key characteristics:

- Receives the triggers from external stimuli, for example the input feature set instances in parallel and processes the feature set.
- The processes within the AI brain are complex compared to the AI Arc
- AI brain’s action footprint is bigger, can involve other components which can trigger post remedial actions to the action performed by the AI Arc.

AI Core – AI Engine:

AI Core engine encompasses Machine Learning and Deep Learning Algorithms. Most of the times during anomaly situations, post the Spine Arc’s sensing act/decision the AI Engine voluntarily captures additional features to investigate and infer more information and trigger complex processes for remedial actions. This could be re- mediating the actions triggered by the AI Arc.

Considering an example from humans it could be the brain triggering the eyes to capture more inputs for further investigation and actions when the hand was placed on a hot surface. The Spinal Arc by then would have triggered withdrawal of the
hands which the brain gets to know but can trigger further features capture and decide on actions.

Considering the pump example earlier, AI Core would receive the decisions made by the AI Arc pertaining to the pump that the pump was shut because it was hot. In order to investigate further and trigger broader actions, AI Brain would trigger capturing more features from the fan or features from inlet valve, runs models on this features set and takes wider actions.

**Habitual Identification Unit (HIU):**

Habitual Identification Unit identifies repetitive feature sets mapped to the value ranges and actions associated to it. HIU gets built over a period of time and is causally related to actions performed by AI BRAIN. At a defined comfort level HIU pushes the feature set/ value range, associated actions to the Reflex Arc (s). Actions associated with HIU can involve a set of action foot prints in a sequence.

**Distributed AI and Reflex – Key differentiators:**

The key difference between traditional distributed functions and Reflex based distributed functions are as follows:

- Reflex based solutions take decisions locally and act upon anomalies
- Actions performed by Reflex actions for anomalies are faster compared to normal distributed mode of functioning
- Reflex based solutions get augmented with Habitual intelligence
- Reflex based solutions may alter the decisions for known anomalies based on the previous experience or sourcing from other knowledge base

**IV. Heuristics**

We created the TIU unit to demonstrate instinct intelligence extracting from factory specifications for an industry pump as follows:
We created a Pilot knowledge base to demonstrate the acquired skill through Natural Language Processing (NLP) algorithms detailed later. The knowledge engine did polling based acquisition.

We used “Elastic search” Engine feature for the following scenario published as Natural representation:

“Pump_1001” was able to give a throughput of 6 litres with Hydrochloric acid of 62% percentage concentration. The propeller depth was 30 meters. At 80 degree temperature the Acid started to evaporate and the pump head had to be increased. Cooling valve had to be triggered off beyond 90 degree to maintain the throughput”.

We extended the depth of the head with CH3COOH to 50 metres and observed the throughput as 7 litres.

**TIU and Action foot print Logic**

Figure below shows the logic for building the TIU and Action foot print. We used the OCR based algorithms in open source Python libraries to extract the name value from factory table, transformed to JSON document format and stored in MongoDB document Data base. This formed the framework for creating TIU table.

From the simulated operation guide we created the Action foot print table. For this we leveraged the elastic search capability to perform matches to entities.

### Pump specifications

| Liquid       | Concentration (%) | Temp Min (deg C) | Temp Max (deg C) | Spec Weight (Kg) | Rated Power Kw | Rated Power HP | Q DELIVERY |
|--------------|-------------------|------------------|------------------|------------------|----------------|----------------|-------------|
| Acetic Acid  | 80                | -10              | 70               | 0.05             | 0.37           | 0.5            | Head metres |
|              |                   |                  |                  |                  |                |                | (within liquid) |

![Figure 6: Pump specifications](image-url)
We then leveraged the preprocessing pipeline to perform fragments of sentences and identified matching sentences. We then fed to the neural parser to extract components dynamically and stored in MongoDB along with conditions. We as well had to supplement the neural parser outputs with regex match. We then performed a match with entities in TIU table and built the Action foot print table as shown in the logical view below.

**Figure 7: TIU and action foot print logic**

We leveraged the below logic for “Acquired Intelligence – TIU” and related Action foot print table.

**Figure 8: TIU table and action foot print**

We leveraged the below logic for “Acquired Intelligence – TIU” and related Action foot print table.
Figure 9: TIU and acquired footprint logic

Logical view of the populated table is shown below:

| Action Domain - TIU - 1 | In contact Liquid | Concentration (%) | Temp Min (deg C) | Temp Max (deg C) | Spec Weight (Kg) | Rated Power Kw | Rated Power HP | Head metres (within liquid) |
|--------------------------|-------------------|-------------------|------------------|------------------|------------------|----------------|----------------|----------------------------|
| Acetic Acid              | 80                | -10               | 70               | 1.05             | 0.37             | 0.5            | Q-DELIVERY 12      | 18, 17, 15.7, 13.8, 11.4, 8.4 |
| Hydrochloric Acid        | 0.62              | NIL               | 79.99            | Implied          | Implied          | Implied        | Q-DELIVERY NIL     | Head Meters 30 degree Self Cooling valve External |

Figure 10: TIU table and acquired footprint

We applied refinement rules based algorithms to define the inferred entities for example if the content specified events above 80 degree the rules based algorithms would infer otherwise below 79.9 degree.

The initial precision and recall range was around 42 and 51 respectively. We increased the elastic search capability and preprocessing framework to achieve 63% of precision and 71% of recall.
We simulated anomaly of farther distributed data and pushed to get the emergency mode triggered.

**AI Brain conceptual Design**

We designed the AI brain to simulate the following scenario:

![AI Brain design](image)

The domain under consideration is fragmented to ownership fractals. While AI based techniques can be applied to identify the fractals we propose manual creation in the current phase. We define ownership fractal as collection of components which can be controlled by the local device; additionally components within each fractal can be controlled by AI brain too. *(Note: Components within the fractal may not have the ability to be auto controlled i.e. need manual intervention).*

Figure above shows 3 fractals.

- Boundary 1 is purely automatic
- Boundary 2 is purely manual
- Boundary 3 is a combination of auto and manual components

We as well manually defined the AI brain table. Each component along with its attributes is encompassed in the table. Communication between the ARC and AI can be either through Pub-Sub model or trap based notifications.
In the below example when the ARC identifies “external domain” entities, it notifies AI. AI subsumes this, investigates and maps it to the relevant fractal boundary.

**Habitual Identification Unit conceptual Design**

Certain actions may need AI interventions. Few example scenarios are below:

- AI Engine possesses better precision and recall compared to ARC
- ARC’s local device initially did not have the ability to perform the relevant control and device upgradation happened at later stages.

For frequently repeating set of features, AI Engine starts to build up the confidence through a “lateral model” by assessing the decisions taken by the local device against itself for the given set of features. (*Note: ARC specific device should be “lateral mode” sensitive and not perform real actions rather send results to AI engine when asked for. Frequency of repeated set of features may be defined through a rules based configuration threshold*).

As time lapses and AI engine senses if ARC is able to take decisions in par with its own, it delegates the decision making ability to the ARC.

**V. Future Work**

We intend to refine the TIU Action footprint logic and develop the AI Engine and HIU in future.
VI. Related Work

D Verma and G Bent [III] specify distributed AI through caching mechanisms. This work specifies the shallow model of AI and depending upon the confidence level of the response the necessary policies get invoked. Our work is a broader one compared to this and focusses on the key types of AI entities.

B. Thuraisingham, J Larson [II] specify the AI based techniques for front end systems in distributed environment. Our work compliments with this for various external AI interfaces.

VI. Conclusion

In this paper we discussed the solution components of Reflex Layer. We also discussed the Anomaly based Reflex and Habitual Reflex. We as well produced the heuristics and the contextual design for AI Brain and HIU. We intend to elaborate the solution for the same as our future work.

(Disclosure: Few of the images used in the figures were sourced from google images)

References

I. Anna Melamed. “Distributed Systems Management on Wall Street- AI Technology needs”

II. B Thuraisingham, J Larson “AI applications in Distributed System Design issues”

III. D Verma, G Bent “Policy Enabled Caching for Distributed AF”. 2017 IEEE International Conference on Big Data (BIGDATA)

IV. Du-Mim Yoon, Joo-Seon Lee, Hyun-SuSeon, Jeong-Hyeon Kim, Kyung-Joong Kim*. “Optimization of Angry Birds AI Controllers with Distributed Computing”. IEEE CIG 2015, Tainan, Taiwan August 31, 2015 – September 2, 2015

V. E Oyekanlu, K Scoles “Towards Low-Cost, Real-Time, Distributed Signal and Data Processing for Artificial Intelligence Applications at Edges of Large Industrial and Internet Networks”. 2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)
VI. Gandhi Sivakumar, G. Arumugam, “The Facets of Distributed And Artificial Intelligence”, Volume8, Issue 12, December 2019 Edition, International Journal of Scientific & Technology Research

VII. JSR Jang, “Self-learning fuzzy controllers based on temporal back propagation”

VIII. Sean Martin, Andrew slade “A methodology for distributed AI and its applicability for Data fusion applications”

IX. YAO Ping-xiang, FENG Wen-xian “DESIGN OF DISTRIBUTED MARINE AISCOMMUNICATION MECHANISM AND INTEGRATED MONITORING PLATFORM”

X. Z. A. Shah1, T. AbdulQayyum2, Dr. S. I. A. Shah “A Distributed Approach for Solving the Position Inverse Kinematics of Mobile Parallel Manipulators”, 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE) Islamabad, Pakistan, April 22-24, 2014