University of St Andrews

University of Dundee

Doctoral Thesis

Rapid Information Factory

Author:
Mr Andreas F. Vermeulen

(1) (2) (3)

Supervisor One:
Dr Vladimir Janjic

Supervisor Two:
Mr Andrew Cobley

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

PhD Research Group
School of Computer Science

August 2015

Declaration of Authorship

Candidate's declarations:

I, Mr Andreas F. Vermeulen, hereby certify that this thesis titled, 'Rapid Information Factory' which is approximately ..... words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in [month, year] and as a candidate for the degree of Doctor of Philosophy in [month, year]; the higher study for which this is a record was carried out in the university of St Andrews between 2015 and [year].

Signed: ______________________________

Date: ______________________________

Supervisor's declaration:

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Doctor of Philosophy in the university of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Signed: ______________________________

Date: ______________________________

Permission for publication:

In submitting this thesis to the university of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the university Library for the time being in force, subject to any copyright vested in the work not being affected thereby.

I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

PRINTED COPY No embargo on print copy

ELECTRONIC COPY No embargo on electronic copy

Signature of candidate:

Date:

Signature of supervisor:

Date:

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITY OF ST ANDREWS

Abstract

Computer Science
School of Computer Science

Doctor of Philosophy

Rapid Information Factory

by Mr Andreas F. VERMEULEN

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

Abbreviations

LAH **L**ist **A**bbreviations **H**ere

Physical Constants

Speed of Light $c = 2.997\ 924\ 58 \times 10^{8}\ \mathrm{ms}^{-\mathrm{S}}$ (exact)

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	$rads^{-1}$

For/Dedicated to/To my…

Chapter 1

Research Summary

1.1 Introduction

Welcome

Chapter 2

Research Question

2.1 Main Section 1

Can a Rapid Information Factory using agile and lean six sigma manufactory principles to solve the issues generated by effective and efficient exascale heterogeneous computing of a quintillion bytes data lake into a value-add deep learning knowledge source?

2.1.1 Subsection 1

???

2.1.2 Subsection 2

???

2.2 Main Section 2

???

Chapter 3

Rapid Information Factory (RIF)

3.1 Main Section 1

???

3.1.1 Subsection 1

???

3.1.2 Subsection 2

???

3.2 Main Section 2

???

Chapter 4

Rapid Information Factory Framework (RIFF)

4.1 Main Section 1

The rapid information factory framework is a methodology, the result of research since 2008 , designed to guide a exascale [1] heterogeneous computing cluster to process a exabyte data lake. The framework will processes a quintillion calculations per second against quintillion bytes of disk storage. The framework generates a series of virtual factories that together process the data lake using enhanced custom designed parallel processes.

4.1.1 Subsection 1

???

4.1.2 Subsection 2

???

4.2 Main Section 2

???

Chapter 5

Rapid Information Factory Framework (RIFF) - Functional Layer

5.1 Main Section 1

The functional layer handles the all functional processes within the cluster. The functions is build to empower the factory to process data sources in a predictable and repeatable series of processes. This layer is the bulk of the framework, as it contains the main components of the factory process and will expand as the factory deploys into full production.

5.1.1 Subsection 1

???

5.1.2 Subsection 2

???

5.2 Main Section 2

???

Chapter 6

Rapid Information Factory Framework (RIFF) - Functional Layer - High-Level View

6.1 Main Section 1

The high-level view of the Homogeneous Ontology for Recursive Uniform Schema (HORUS) shows the users the current status of the rapid information factory. This is achieve by visualisation of the rapid information factory via a Rstudio Shiny [2] and R [3] based web site. The complete state of the factories are online in a graph database.

6.1.1 Subsection 1

???

6.1.2 Subsection 2

???

6.2 Main Section 2

???

Chapter 7

Rapid Information Factory Framework (RIFF) - Functional Layer - Synaptic Assimilator (SA)

7.1 Main Section 1

The synaptic assimilator (SA) is an artificial intelligence [4] engine that performs the processes assigned to the system to the most effective and efficient method.

The artificial intelligence uses machine learning as an investigation and testing method to improve and select the correct combination of processing artifacts to achieve the efficient and effective outcome of each factory process.

The engine is taught data science and performance improvement processes to enables it to manage the factories to process the data sources it is supplied with for processing.

7.1.1 Subsection 1

???

7.1.2 Subsection 2

???

7.2 Main Section 2

???

Chapter 8

Rapid Information Factory Framework (RIFF) - Functional Layer - Exascale Data Lake

8.1 Main Section 1

The exascale data lake is a data source that exceeds a quintillion bytes. The data source holds structured, semi-structured and unstructured data. The synaptic assimilator converts it into a base deep learning data source by applying the appropriate functions and data processing patterns.

8.1.1 Subsection 1

???

8.1.2 Subsection 2

???

8.2 Main Section 2

???

Chapter 9

Rapid Information Factory Framework (RIFF) - Functional Layer - Persistent Recursive Information Schema Manipulator (PRISM)

9.1 Main Section 1

The persistent recursive information schema manipulator is the central control framework for each data procesing flow through the system using a bulk synchronous parallel (BSP) abstract computer as a bridging model for rapid information factory's parallel algorithms, that are pre-defined and tested by each factory.

9.1.1 Subsection 1

???

9.1.2 Subsection 2

???

9.2 Main Section 2

???

Chapter 10

Rapid Information Factory Framework (RIFF) - Functional Layer - RAPTOR Supersteps

10.1 Main Section 1

The RAPTOR framework is the basis for the six supersteps of the bulk synchronous parallel (BSP) based process. The framework uses fundamental building blocks like pipeline, farm and loopback to formulate more complex structures to handle the data requirements within each of the six supersteps.

10.1.1 Subsection 1

???

10.1.2 Subsection 2

???

10.2 Main Section 2

???

Chapter 11

Rapid Information Factory Framework (RIFF) - Functional Layer - Retrieve Superstep

11.1 Main Section 1

The retrieve superstep is responsible for data retrieval from other data sources into the data lake.

11.1.1 Subsection 1

11.1.1.1 Retrieve Superstep

The retrieve superstep uses a series of work cells with an assembly format that is made up out of four components:

- Remote Monitor Yoke
- Input PUPA or Input NEST PUPA
- ANT
- Output PUPA

The remote monitoring yoke connects the work cell to the PRISM that controls the spesific factory to enable the communication and control to the PRISM's remote motering yoke to ensure the process is monitored and that it complies to its assigned task in the factory.

The Input PUPA or Input NEST PUPA describes the processsing rules and formats of the data source use as the input to the process. The factory translates the HORUS scripts to generate the processing logic to input the data.

The ANT is the setup script in HORUS rules that builds a processing engine to process the data form the input PUPA into the processing rules of the output PUPA.

The output PUPA is the processsing rules and formats of the data source use as the output from the process. The factory translates the HORUS scripts to generate the processing logic to output the data.

11.1.2 Subsection 2

???

11.2 Main Section 2

???

Chapter 12

Rapid Information Factory Framework (RIFF) - Functional Layer - Assess Superstep

12.1 Main Section 1

The assess superstep is responsible for the data validation in the active factory. Data Quality is determined and also improved where possible in this super step.

12.1.1 Subsection 1

12.1.1.1 Assess Superstep

The assess superstep uses a series of work cells with an assembly format that is made up out of four components:

- Remote Monitor Yoke
- Input PUPA
- ANT
- Output PUPA

The remote monitoring yoke connects the work cell to the PRISM that controls the spesific factory to enable the communication and control to the PRISM's remote motering yoke to ensure the process is monitored and that it comples it assigned task in the factory.

The Input PUPA NEST PUPA describes the processsing rules and formats of the data source use as the input to the process. The factory translates the HORUS scripts to generate the processing logic to input the data.

The ANT is the setup script in HORUS rules that builds a processing engine to process the data form the input PUPA into the processing rules of the output PUPA.

The output PUPA is the processsing rules and formats of the data source use as the output from the process. The factory translates the HORUS scripts to generate the processing logic to output the data.

12.1.2 Subsection 2

???

12.2 Main Section 2

???

Chapter 13

Rapid Information Factory Framework (RIFF) - Functional Layer - Process Superstep

13.1 Main Section 1

The process superstep is responsible for the processing of the dark data in the data lake into a structured data vault. The data vault keeps full record of the different phases that the data is process over time.

13.1.1 Subsection 1

13.1.1.1 Process Superstep

The process superstep uses a series of work cells with an assembly format that is made up out of four components:

- Remote Monitor Yoke
- Input PUPA
- ANT
- Output PUPA

The remote monitoring yoke connects the work cell to the PRISM that controls the spesific factory to enable the communication and control to the PRISM's remote motering yoke to ensure the process is monitored and that it comples it assigned task in the factory.

The Input PUPA NEST PUPA describes the processsing rules and formats of the data source use as the input to the process. The factory translates the HORUS scripts to generate the processing logic to input the data.

The ANT is the setup script in HORUS rules that builds a processing engine to process the data form the input PUPA into the processing rules of the output PUPA.

The output PUPA is the processsing rules and formats of the data source use as the output from the process. The factory translates the HORUS scripts to generate the processing logic to output the data.

13.1.2 Subsection 2

???

13.2 Main Section 2

???

Chapter 14

Rapid Information Factory Framework (RIFF) - Functional Layer - Transform Superstep

14.1 Main Section 1

The transform superstep is responsible for transforming the data vault into an enterprise data warehouse. This superstep tranforms the data into knowledge by adding dimensionality and insight to the data vault.

14.1.1 Subsection 1

14.1.1.1 Transform Superstep

The transform superstep uses a series of work cells with an assembly format that is made up out of four components:

- Remote Monitor Yoke
- Input PUPA
- ANT
- Output PUPA

The remote monitoring yoke connects the work cell to the PRISM that controls the spesific factory to enable the communication and control to the PRISM's remote motering yoke to ensure the process is monitored and that it comples it assigned task in the factory.

The Input PUPA NEST PUPA describes the processsing rules and formats of the data source use as the input to the process. The factory translates the HORUS scripts to generate the processing logic to input the data.

The ANT is the setup script in HORUS rules that builds a processing engine to process the data form the input PUPA into the processing rules of the output PUPA.

The output PUPA is the processsing rules and formats of the data source use as the output from the process. The factory translates the HORUS scripts to generate the processing logic to output the data.

14.1.2 Subsection 2

???

14.2 Main Section 2

???

Chapter 15

Rapid Information Factory Framework (RIFF) - Functional Layer - Organise Superstep

15.1 Main Section 1

The organise superstep is responsible for organising the data sets together for each business grouping from the data warehose into data marts.

15.1.1 Subsection 1

15.1.1.1 Organise Superstep

The organise superstep uses a series of work cells with an assembly format that is made up out of four components:

- Remote Monitor Yoke
- Input PUPA
- ANT
- Output PUPA

The remote monitoring yoke connects the work cell to the PRISM that controls the spesific factory to enable the communication and control to the PRISM's remote motering yoke to ensure the process is monitored and that it comples it assigned task in the factory.

The Input PUPA NEST PUPA describes the processsing rules and formats of the data source use as the input to the process. The factory translates the HORUS scripts to generate the processing logic to input the data.

The ANT is the setup script in HORUS rules that builds a processing engine to process the data form the input PUPA into the processing rules of the output PUPA.

The output PUPA is the processsing rules and formats of the data source use as the output from the process. The factory translates the HORUS scripts to generate the processing logic to output the data.

15.1.2 Subsection 2

???

15.2 Main Section 2

???

Chapter 16

Rapid Information Factory Framework (RIFF) - Functional Layer - Report Superstep

16.1 Main Section 1

The report superstop is responsible to perform the required reporting and analytic requirements.

16.1.1 Subsection 1

16.1.1.1 Report Superstep

The report superstep uses a series of work cells with an assembly format that is made up out of four components:

- Remote Monitor Yoke
- Input PUPA
- ANT
- Output PUPA

The remote monitoring yoke connects the work cell to the PRISM that controls the spesific factory to enable the communication and control to the PRISM's remote motering yoke to ensure the process is monitored and that it comples it assigned task in the factory.

The Input PUPA NEST PUPA describes the processsing rules and formats of the data source use as the input to the process. The factory translates the HORUS scripts to generate the processing logic to input the data.

The ANT is the setup script in HORUS rules that builds a processing engine to process the data form the input PUPA into the processing rules of the output PUPA.

The output PUPA is the processsing rules and formats of the data source use as the output from the process. The factory translates the HORUS scripts to generate the processing logic to output the data.

16.1.2 Subsection 2

???

16.2 Main Section 2

???

Chapter 17

Rapid Information Factory Framework (RIFF) - Operational Management Layer

17.1 Main Section 1

The operational mangement layer is responsible to handle all the arctifacts required by the factory to perform the required processing requirements. It also bounds the factory to only be able to perform functions the layer already manages.

17.1.1 Subsection 1

???

17.1.2 Subsection 2

???

17.2 Main Section 2

???

Chapter 18

Rapid Information Factory Framework (RIFF) - Autonomous Node Transport (ANT) Definitions

18.1 Main Section 1

The autonomous node transport definitions are the set of cloud instances or physical servers configurations supporting the processing capability of the factory. The HORUS schema keeps a series of characteristics required by the factory to decide which autonomous node transport to use for which requirement. The nodes will be changed in the future as characteristics for new nodes are discovered and loaded into the factory.

The nodes are heterogeneous computing enabled and support combinations of heterogeneous processors that covers the range from high-end servers and high-performance computing machines to low-power embedded devices like mobile phones and tablets.

18.1.1 Subsection 1

???

18.1.2 Subsection 2

???

18.2 Main Section 2

???

Chapter 19

Rapid Information Factory Framework (RIFF) - Autonomous Node Transport Management (ANTM)

19.1 Main Section 1

The autonomous node transport management oversees the complete process of running the heterogeneous computing systems. The combinations and work flow of the nodes is pre-defined in this section.

19.1.1 Subsection 1

???

19.1.2 Subsection 2

???

19.2 Main Section 2

???

Chapter 20

Rapid Information Factory Framework (RIFF) - Monitoring

20.1 Main Section 1

Monitoring handles the monitoring tasks in the system. The monitoring covers all aspects of the factory's performance.

20.1.1 Subsection 1

???

20.1.2 Subsection 2

???

20.2 Main Section 2

???

Chapter 21

Rapid Information Factory Framework (RIFF) - Persistent Uniform Protocol Agreement (PUPA) Definitions

21.1 Main Section 1

The persistent uniform protocol agreement definitions are the collection of the algorithmic skeletons within the system.The PUPAs are programs generate using existing frameworks like OpenCL [5], ArrayFire [6], Spark [7], Titan graph database [8] [9], FastFlow Framework [10]. This is the main area of research for the team. The enhancement and creation of new parallel patterns will improve the capasity of the synaptic assimilator to build new factories and process the data into knowledge.

21.1.1 Subsection 1

???

21.1.2 Subsection 2

???

21.2 Main Section 2

???

Chapter 22

Rapid Information Factory Framework (RIFF) - Persistent Uniform Protocol Agreement (PUPA) Management

22.1 Main Section 1

The persistent uniform protocol agreement management oversees the complete collection. The process ensures that all uniform protocol agreements are manange to achieve the required end goal of the factory to effectively and efficently process data into knowledge.

22.1.1 Subsection 1

???

22.1.2 Subsection 2

???

22.2 Main Section 2

???

Chapter 23

Rapid Information Factory Framework (RIFF) - Alerting

23.1 Main Section 1

The alerts are manage from this singular point in the system. The alerting process interacts with the communication process to ensure the appropriate response is generated for the alerts.

23.1.1 Subsection 1

???

23.1.2 Subsection 2

???

23.2 Main Section 2

???

Chapter 24

Rapid Information Factory Framework (RIFF) - Parameters

24.1 Main Section 1

The parameters are stored in this singular place in the system. Managing the parameters in a single location enhance sthe factory to adapt to changing requirements in a rapid time.

24.1.1 Subsection 1

???

24.1.2 Subsection 2

???

24.2 Main Section 2

???

Chapter 25

Rapid Information Factory Framework (RIFF) - Scheduling

25.1 Main Section 1

The parameters are stored in this singular place in the system. Managing the parameters in a single location enhance sthe factory to adapt to changing requirements in a rapid time.

25.1.1 Subsection 1

???

25.1.2 Subsection 2

???

25.2 Main Section 2

???

Chapter 26

Rapid Information Factory Framework (RIFF) - Communication

26.1 Main Section 1

Communication handles the communication into and from the system from this singular point.

26.1.1 Subsection 1

???

26.1.2 Subsection 2

???

26.2 Main Section 2

???

Chapter 27

Rapid Information Factory Framework (RIFF) - Audit, Balance and Control (ABC) Layer

27.1 Main Section 1

Audit, Balance and Control (ABC) Layer

27.1.1 Subsection 1

???

27.1.2 Subsection 2

???

27.2 Main Section 2

???

Chapter 28

Rapid Information Factory Framework (RIFF) - Audit, Balance and Control (ABC) Layer - Work Cells

28.1 Main Section 1

The work cells [11] is the fundamental building block of the processing system.

The work cell is executing as an actor model (a mathematical model of concurrent computation) that use "actors" as the universal primitives of concurrent computation: in response to a message it receives, an actor can make local decisions, create more actors, send more messages, and determine how to respond to message received.

28.1.1 Work Cells

The remote work cells is the fundamental processing container of the rapid information factory.

28.1.1.1 Monitor Work Cell

The monitor work cell consists of a persistent recursive information schema manipulator plus a remote assessment yoke for each processing work cell the spesific BSP flow requires in the rapid information factory.

28.1.1.2 Processing Work Cell

The processing work cell is a combination of a remote assessment yoke, an input persistent uniform protocol agreement, an autonomous node transport and an output persistent uniform protocol agreement. The remote assessment yoke communicates to the remote assessment yoke attached to the monitor work cell. The input persistent uniform protocol agreement holds the instructions to enable the work cell to import the data into the work cell. The output persistent uniform protocol agreement holds the instructions to enable the work cell to export the data from the work cell.The autonomous node transport supplies the processing power to execute the PUPA and the yoke instructions.

28.1.1.3 Measure Work Cell

The measure work cell consists of an autonomous node transport that supplies the processing power and a measure agreement precision that supplies the tests to determine if the processing was successful.

28.1.2 Subsection 2

???

28.2 Main Section 2

???

Chapter 29

Rapid Information Factory Framework (RIFF) - Audit, Balance and Control (ABC) Layer - Execution Statistics

29.1 Main Section 1

he execution statistics is the fundamental performance recording system for the factory in the solution.

29.1.1 Subsection 1

???

29.1.2 Subsection 2

???

29.2 Main Section 2

???

Chapter 30

Rapid Information Factory Framework (RIFF) - Audit, Balance and Control (ABC) Layer - Remote Yoke

30.1 Main Section 1

The Yoke or Poka-yoke is fundamental process of ”mistake-proofing”. The Remote York is the rapid information factory’s fundamental monitoring interface between the different work cells.

30.1.1 Subsection 1

???

30.1.2 Subsection 2

???

30.2 Main Section 2

???

Chapter 31

Rapid Information Factory Framework (RIFF) - Audit, Balance and Control (ABC) Layer - Rejections and Error Handling

31.1 Main Section 1

The rejections and error handling in the rapid information factory handles the rejections and error handling within the system.

31.1.1 Subsection 1

???

31.1.2 Subsection 2

???

31.2 Main Section 2

???

Chapter 32

Rapid Information Factory Framework (RIFF) - Audit, Balance and Control (ABC) Layer - Balancing and Control

32.1 Main Section 1

The balance and control mechanisms are the execute from the singular section.

32.1.1 Subsection 1

???

32.1.2 Subsection 2

???

32.2 Main Section 2

???

Chapter 33

Rapid Information Factory Framework (RIFF) - Audit, Balance and Control (ABC) Layer - Codes Management

33.1 Main Section 1

The code management is the single section that holds the standard codes used in the system.

Standard codes includes ISO codes, pre-agreed names and known lists of items for the factory.

The use of standard codes enable the effective deep data mining prescribed as output of the factory.

33.1.1 Subsection 1

Example of the ISO standards used are:

- ISO 3166 - Country Codes of the world
- ISO 4217 - Currency Codes of the world
- ISO 6709 - Representation of latitude, longitude and altitude for geographic point locations.

- ISO 639 - Language codes
- ISO 19134:2006 - Multimodal location-based services for routing and navigation.
- ISO 4030:1983 - Vehicle identification number (VIN)

33.1.2 Subsection 2

???

33.2 Main Section 2

???

Chapter 34

Rapid Information Factory Framework (RIFF) - Business Layer

34.1 Main Section 1

Business Layer

34.1.1 Subsection 1

???

34.1.2 Subsection 2

???

34.2 Main Section 2

???

Chapter 35

Rapid Information Factory Framework (RIFF) - Business Layer - Functional Requirements

35.1 Main Section 1

A functional requirement defines a function of a system and its components. A function is described as a set of inputs, the behavior, and outputs. [12]. The set of requirements together as a unit describes the factory processing rules.

35.1.1 Subsection 1

???

35.1.2 Subsection 2

???

35.2 Main Section 2

???

Chapter 36

Rapid Information Factory Framework (RIFF) - Business Layer - Non-Functional Requirements

36.1 Main Section 1

A non-functional requirement is a requirement that specifies criteria that can be used to test the operation of a factory, rather than specific behaviors of the process within the factory.The set of requirements together as a unit describes the factory verification rules.

36.1.1 Subsection 1

The following are types of non-functional requirements:

- Accessibility
- Audit and control
- Availability

  Availability of factory as a factor of its reliability.

- Backup
- Capacity (current and forecast)
- Certification

  Certification of the factory can be achieved for several different ISO standards.
- Compliance

  Compliance is achieved against a minimum set of criteria for the factory.
- Configuration management
- Dependency on other parties
- Deployment
- Documentation
- Disaster Recovery
- Efficiency
- Effectivenes
- Emotional factors
- Environmental protection
- Escrow
- Exploitability
- Extensibility

- Failure management
- Fault tolerance
- Legal and licensing issues
- Patent-infringement-avoidability
- Interoperability
- Maintainability
- Modifiability
- Network topology
- Open source
- Operability
- Performance / Response time

  Short response time for a spesific PUPA to complete.

  High throughput in the factory

  Low utilization of computing resources in the factory.

- Platform compatibility
- Price/Cost
- Privacy
- Portability
- Quality

- Recovery / Recoverability
- Reliability
- Reporting
- Resilience
- Resource constraints
- Response time

  Requirement for timely response from the factory.

- Reusability
- Robustness
- Safety
- Scalability

  The horizontal and vertical scalability is the ratio the factory can expand its resources for processing.

- Security
- Software tools
- Stability
- Standards
- Supportability
- Testability

- Usability by user community
- User Friendliness

36.1.2 Subsection 2

???

36.2 Main Section 2

???

Chapter 37

Rapid Information Factory Framework (RIFF) - Utility Layer

37.1 Main Section 1

The utility layer stores processing structures across the factory for general or common requirements.

37.1.1 Subsection 1

???

37.1.2 Subsection 2

???

37.2 Main Section 2

???

Chapter 38

Rapid Information Factory Framework (RIFF) - Maintenance Utilities

38.1 Main Section 1

The maintenance untilities are procesing structures that perform work for the factory to maintain.

38.1.1 Subsection 1

???

38.1.2 Subsection 2

???

38.2 Main Section 2

???

Chapter 39

Rapid Information Factory Framework (RIFF) - Data Utilities

39.1 Main Section 1

Data Utilities are procesing structures that perform work for the factory to maintain data.

39.1.1 Subsection 1

???

39.1.2 Subsection 2

???

39.2 Main Section 2

???

Chapter 40

Rapid Information Factory Framework (RIFF) - Spesific Utilities

40.1 Main Section 1

Spesific Utilities.

40.1.1 Subsection 1

???

40.1.2 Subsection 2

???

40.2 Main Section 2

???

Chapter 41

Rapid Information Factory Framework (RIFF) - Spesific Utilities - Autonomous Logical Agreement Transport Executor (ALATE)

41.1 Main Section 1

The autonomous logical agreement transport executor is a special utility that builds a fundamental metadata view of the data source it is processing and configures a fundamental set of PUPA that will form the fundamental factory for the data source.

41.1.1 Subsection 1

???

41.1.2 Subsection 2

???

41.2 Main Section 2

???

Chapter 42

Rapid Information Factory Framework (RIFF) - Spesific Utilities - Rapid Artifical Intelligence Data Extract Routine (RAIDER)

42.1 Main Section 1

The rapid artifical intelligence data extract routine is a special utility that builds process PUPA for Retrieve of data via NEST PUPA.

42.1.1 Subsection 1

???

42.1.2 Subsection 2

???

42.2 Main Section 2

???

Chapter 43

Rapid Information Factory Framework (RIFF) - Spesific Utilities - Rapid Execute Artificial Protocol Engine for Routine (REAPER)

43.1 Main Section 1

The rapid execute artificial protocol engine for routine is a special utility that performs a stand alone execution of any pre-approved work cell.

43.1.1 Subsection 1

???

43.1.2 Subsection 2

???

43.2 Main Section 2

???

Chapter 44

Rapid Information Factory Framework (RIFF) - Spesific Utilities - Sequencetial Converter into Ontology for Uniform Transport (SCOUT)

44.1 Main Section 1

The sequencetial converter into ontology for uniform transport is a special utility that builds NEST PUPA for a data source.

The SCOUT connects to the external data and discovers the metadata required to connect to the data source.

The output is a NEST script using HORUS scripts to be used by the factory to connect to the spesific data source.

44.1.1 Subsection 1

???

44.1.2 Subsection 2

???

44.2 Main Section 2

???

Chapter 45

Rapid Information Factory Framework (RIFF) - Rapid Information Factory Data Sources

45.1 Main Section 1

Rapid Information Factory Data Sources

45.1.1 Subsection 1

???

45.1.2 Subsection 2

???

45.2 Main Section 2

???

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

[1] Keren Bergman, Shekhar Borkar, Dan Campbell, William Carlson, William Dally, Monty Denneau, Paul Franzon, William Harrod, Kerry Hill, Jon Hiller, et al. Exascale computing study: Technology challenges in achieving exascale systems. *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep*, 15, 2008.

[2] Felipe Ortega, Javier M Moguerza, and Emilio L Cano. Combining r and python for scientific computing. In *The R User Conference, useR! 2013 July 10-12 2013 University of Castilla-La Mancha, Albacete, Spain*, volume 10, page 92, 2013.

[3] R Core Team. R language definition, 2000.

[4] Daniel E O'Leary. Artificial intelligence and big data. *IEEE Intelligent Systems*, (2):96–99, 2013.

[5] John E Stone, David Gohara, and Guochun Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering*, 12(1-3):66–73, 2010.

[6] James Malcolm, Pavan Yalamanchili, Chris McClanahan, Vishwanath Venugopalakrishnan, Krunal Patel, and John Melonakos. Arrayfire: a gpu acceleration platform. In *SPIE Defense, Security, and Sensing*, pages 84030A–84030A. International Society for Optics and Photonics, 2012.

[7] Pieter Hintjens. Ømq-the guide. *Online: http://zguide. zeromq. org/page: all, Accessed on*, 23, 2011.

[8] Ilie Tanase, Yinglong Xia, Lifeng Nai, Yanbin Liu, Wei Tan, Jason Crawford, and Ching-Yung Lin. A highly efficient runtime and graph library for large scale graph analytics. In *Proceedings of Workshop on GRAph Data management Experiences and Systems*, pages 1–6. ACM, 2014.

[9] Vivek Mishra. Titan graph databases with cassandra. In *Beginning Apache Cassandra Development*, pages 123–151. Springer, 2014.

[10] Marco Aldinucci, Marco Danelutto, Peter Kilpatrick, Massimiliano Meneghin, and Massimo Torquati. Accelerating code on multi-cores with fastflow. In *Euro-Par 2011 Parallel Processing*, pages 170–181. Springer, 2011.

[11] William M Feld. *Lean manufacturing: tools, techniques, and how to use them.* CRC Press, 2000.

[12] Gruia-Catalin Roman. A taxonomy of current issues in requirements engineering. *Computer*, 18(4):14–23, 1985.