# Agent Based Virtual Research Assistants for E-Science Infrastructures


By
Muhammad Usama
2011-NUST-MS-CS-009

Supervisor
Dr. Peter Bloodsworth
Department of Computing

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Computer Science (MSCS)

In
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(July 2015)

# CERTIfiCATE OF ORIGINALITY

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Muhammad Usama**

Signature: _____

# ACKNOWLEDGMENTS

First of all I want to thank my parents for their full support and everything they did to help and encourage me. Especially to my sister and brother who motivated me to accomplish this goal.

I would like to thank specially to my Supervisor Dr. Peter Bloodsworth for his guidance, useful input and availability throughout my research work. His support and encouragement was remarkable and responded to my questions and queries so promptly. I have to appreciate his enthusiasm and instant review of my thesis which able me to complete my research work.

I would like to express my deep gratitude to Dr. Sheeba Murad for her throughout help in the case study that I used. I would also like to thank Dr. Muhammad Sohail Iqbal, Mr. Shamyl Bin Mansoor for being my committee members. I would also like to thank Dr. Ali Mustafa Qamar for assisting me during my time as their student.

I would also extend my thanks to my friends Nauman Khalid, Iqbal Aziz, Tariq Habib Afridi and Ali Shahzad and Sheraz Anjum for their help at different stages, discussions and valuable feedback. Their discussion and motivations were always useful for me. They motivated me during the tough time in thesis journey. I would like to thank everyone who is directly or indirectly contributed in my thesis work.

# ABSTRACT

Currently scientists around the world are working to make new discoveries and advancements using e-Science platforms. Modern science relies on computational power and uses large data which often requires grid or utility computing. There are algorithms that analyse this large-scale data using significant quantities of computational power. Terabytes of data need to be mined for these algorithms to work effectively. e-Science provides scientists environment to work on this data using GUIs, workflow engines and other tools. The volumes of data that are becoming available for use in e-Science is growing exponentially and it will be very difficult in near future to process all of it in order to carry out research. It will therefore become extremely easy to miss important data which might yield useful findings. This thesis presents an approach which uses a MAS (Multi-agent Systems) to assist the researcher in carrying out their work and to manage large-scale data effectively. With current approaches the user runs a scientific workflow and then has to wait possibly several hours for output. They will need to repeat the entire process again if the results are not accurate, either by using new data or by changing algorithm parameters. Our proposed approach provides user a fairly generic way to automate this process. The user provides the system with the workflow and the kind of results that they are interested in using an intuitive interface. The MAS then goes through the data, selects the samples and passes them to the workflow execution environment. The user is notified of the desired results whenever they become available. In order to evaluate this approach a prototype has been implemented on a real-world bioinformatics case study. Experimental results have found that a multi-agent system can assist the user by saving both their time and effort in the analysis of large data sets. A researcher can assign agents to do basic repetitive tasks on their behalf which frees up their time and energy for more productive tasks.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1 - INTRODUCTION

## 1.1 INTRODUCTION

E-Science is becoming increasingly important to the advancement of modern science. It is computationally intensive and uses often large datasets that may require grid computing. Cluster, grid or cloud computing infrastructures are commonly utilised to process complex large-scale data sets in the search for new knowledge. There are algorithms that analyse this amount of data using a significant amount of computational power. Terabytes of data need to be mined for these algorithms to work. E-Science platforms commonly provide scientists with a research environment in which they can conveniently work with potentially terabytes of data using graphical interfaces and workflow engines. E-Science is being used in particle physics, bio-informatics, earth sciences and social simulations. As the software is commonly complex and data is huge large teams of scientists from groups including universities, government bodies and research laboratories collaborate on it. Some examples of successful E-Science platforms are Open Science Grid, neuGRID and the World Wide LHC Computing project.

As the data in many domains is growing at an almost exponential rate, it will become increasingly difficult in the near future to process all of it. According to a report [1] from digital health consultancy DrBonnie360, there is an estimated 500 petabytes of data in the healthcare realm. This is predicted to grow, by a factor of 50, to 25,000 petabytes by 2020. This is 50 times more than current amount of data in just 8 year timespan. With this near speed of growth in data there is a significant chance that the researchers will miss out important data which could delay or prevent significant discoveries from being made.

There needs therefore to be some automated way to do the research without missing anything important. One idea is to eliminate the human from the process and make a

fully automated system which will look for data, analyses it and in the end tell the important results to the user. There has been research going on making artificial scientists [2], [3] that carry out the researches on their own by formulating the problems as well as their solutions. Given the complexity of the research process and the analytical skills that are necessary it is beyond the capabilities of machines to fully automate many fields. Another idea was to use an Intelligent Software Assistant that will help the researcher carrying out his tasks. Instead of doing everything on its own, the intelligent assistant takes the users requirements and goal, and come up with the answer user is interested in. This whilst being a more realistic goal in the medium term, is still full of challenges.

## 1.2 Problem Statement

The data used for research is growing rapidly and in near future researcher may miss out important data which may delay or prevent important discoveries. The purpose of this research is to make an agent based virtual research assistant for E-Science platforms. It will help the researcher by applying suitable workflows on the data that it selects automatically and give the results to the user if they match the criteria defined by user.

**Research Hypothesis**

*"Multi-Agent based virtual research assistants can be used to automate data selection and workflow analysis on E-Science platforms"*

Virtual research assistants have been seen helping users in other domains and technologies. Currently there is not a generic multi-agent based system to assist user in lengthy workflow analysis tasks which waste a lots of researcher's time. Thus there is a need to put forward an approach to help save their time by assisting them.

## 1.3 Research Questions

RQ1: What are the current limitations in data analysis tasks?

It will be seen how the researchers perform the data analysis tasks and what are their limitations. This will be answered in the literature review chapter.

RQ2: How can virtual assistant be designed and made to work?

A multi-agent layer approach has been proposed which interacts which connects user with the underlying system with an interface to assist the user in a helpful way. This question can be answered in Chapter 3.

RQ3: How can they help researchers in automating tasks?

The virtual research assistants should be able to help users in automating the workflow they want to execute as this is the aim of this research. We will see that how will they be made so that they can help the users in Chapter 4.

RQ4: How well multi-agent based virtual research assistant work?

Proposed approach will be tested on a real world case scenario. Experiments will be performed and evaluated on defined metrics to judge how well it works. This question will be answered in Chapter 5.

## 1.4 Thesis Structure

This thesis is structured as following. Research problem is introduced in the beginning of this chapter. Then hypothesis is stated for this research problem. Following hypothesis, research questions are presented with description and methodology for each. Then methodology explains the research problem and the adopted methodology to find the solution with proposed approach. Next section discusses the objectives of other chapters of the thesis.

Chapter two of the thesis presents detailed literature review with background for hypothesis and reveal how important is the research problem. Analysis of other approaches to solve the similar research problem is summarized and discussed with shortcomings of other approaches. It will be shown with the literature review chapter that proposed work has not been done before and it will be useful to work in this area with proposed methodology.

Chapter three presents design and architecture of the proposed multi-agent system. A multi-agent layer system has been proposed and its architecture has been explained in this chapter. The architecture uses a workflow system with which it integrates and

works using an agent. Detailed architecture of each agent with explanation of how it works is also explained in the chapter.

Chapter four details the implementation of the system following the proposed architecture in chapter three. Modules and agents presented in the architecture are implemented. A workflow system has been implemented first. Than a real-world bioinformatics case study is implemented on it so that it can be tested and evaluated.

Chapter five contains the evaluation of the implemented prototype. Experimentation and evaluation are done using bioinformatics case study which has been implemented in chapter 4. Test metrics have been constructed and test cases and experiments are created for comprehensive testing of the prototype. Prototype is evaluated by qualitative and quantitative testing. Results for all conducted experiments are discussed and analysed in the end.

Chapter 6 presents the conclusion of the research and future work on proposed approach. Various research problems which can be carried out on the basis of this thesis are presented.

# Chapter 2 - Literature Review & Background

## 2.1 What is E-Science?

E-Science is $120M UK initiative to help science work with large amount of data in collaboration with researchers from different institutes [4][5]. The term E-Science was introduced by Dr. John Taylor. In his words “e-Science is about global collaboration in key areas of science and the next generation of infrastructure that will enable it.” The e-science platforms are generally used on grid infrastructures. The grid is a large distributed system which is heterogeneous, loosely coupled and geographically dispersed. A grid is commonly used for various purposes but it can also be dedicated to a particular application.

## 2.2 E-Science Platforms

This thesis will take a motivational example from current developments in this e-Science area. neuGRID [6] was selected as a potential application area and case study for the research work because it is an e-Science platform, which is equipped with a user-friendly interface to allow neuroscientists carry out research into degenerative brain diseases. In particular its main focus is Alzheimer’s disease. This platform allows the collection of a large amount of imaging data and combines this with computationally intensive data analysis tools. With neuGRID neuroscientists can identify neurodegenerative disease markers through the analysis of 3D magnetic resonance brain images. This is done via sets of distributed medical and Grid services. It aims to become the “Google for brain imaging” by providing a virtual imaging

laboratory that is accessible with only a computer and web browser. Its architecture is designed in such a way that it can be adapted for generic medical services other than Alzheimer and the neurosciences. neuGRID uses a mix of JAVA and Web technologies to address its end-users requirements. N4U (neuGRID for you) is an expansion of neuGRID services to the new user communities. It uses 3-tier architecture as shown in the image below

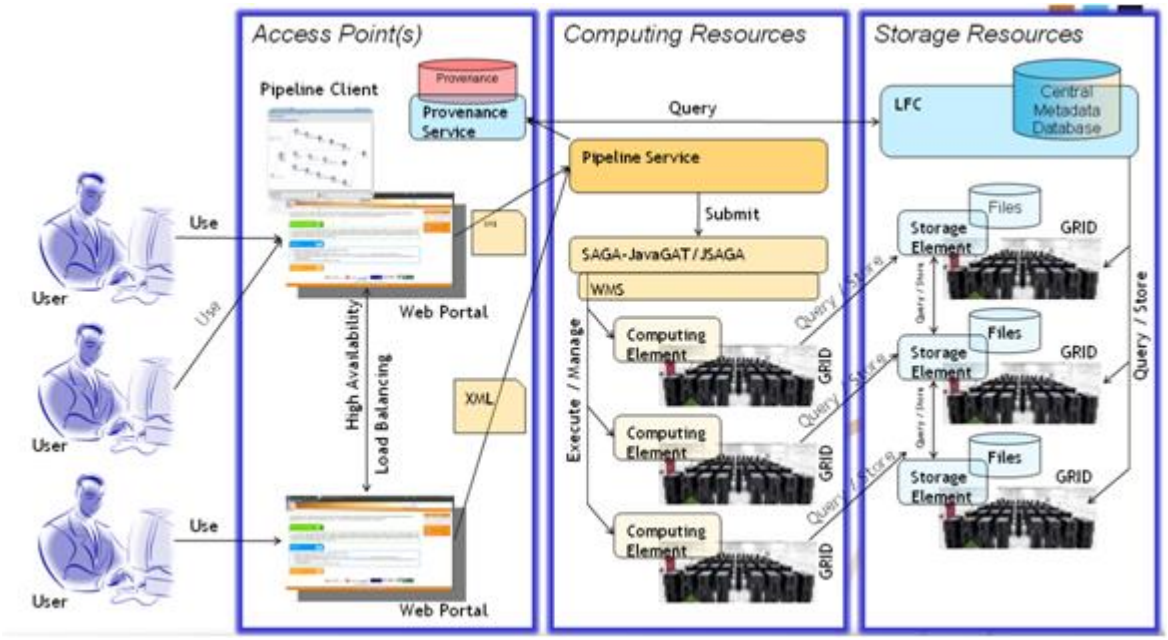


*Figure 2-1 neuGrid N4U 3-tier architecture*

This 3-tier architecture delivers computing and storage resources to the users. One can use neuGRID services with a simple user interface. OutGrid [7] is another platform promoting interoperability among three e-infrastructures named neuGRID, LONI [8] and CBRAIN [9].

Another e-Science platform is myExperiment [10], which facilitates the scientists in collaborating by sharing the research related items and in particular by sharing and executing the workflows [11]. It is the largest public repository of the scientific workflows. All of these systems however rely on the researcher to interact with the system. Since the time of such users is limited it would appear beneficial if some of the more basic or common tasks could be automated in some way.

The virtual Kidney is a platform which provides analysts and experimental scientist with access to knowledge database and computational simulations which are hosted in geographically separated libraries. It uses distributed computing and provides a web interface where users, without requiring a specific programming environment, can explore a range of complex models.

## 2.3 Artificial Scientists:

David Waltz and Bruce G. Buchanan proposed the idea of automating science [9]. According to them it is possible for a computer program to conduct a continuously looping procedure that starts with a question, carry out experiments to answer the question, evaluate the result and reformulate new question. Scientists have been trying to automate the scientific methodology to do experiments and extract new interesting results from some time. IBM's Herbert Gelernter authored a program that rediscovered Euclid's geometry theorems, but it relied too much on rules supplied by the programmer. In the 1970s, Douglas Lenat's Automated Mathematician automatically generated mathematical theorems, but they proved largely useless.

Recently In a research to find equations of natural laws automatically, Lipson and Schmidt made a computer program. The defined an algorithm to find analytics relations automatically. They gave the program the motion tracking data captured from various physical systems. Without any previous knowledge of physics, kinematics or geometry the algorithm discovered Hamiltonians, Lagrangians, and other laws of geometric and momentum conservation [2].

ADAM, a robot scientist can hypothesize and try to solve the problem itself. To test its capabilities it was given the task to discover more about the genome of baker's yeast. After a crash course of biology, it quickly set to work by formulating and testing 20 different hypotheses. The robot eventually identified the genes that code for enzymes involved in yeast metabolism, a scientific first for a robot [3][12].

## 2.4 Virtual Assistants

A fully automated system like ADAM is too complex for our problem. It hypothesize and try to solve the problem all by itself while the aim of this thesis is to solve a particular problem for the researcher. For the E-Science platforms virtual research assistants instead of automating the entire research process they can simply assist the researcher. Instead of doing everything on its own, this intelligent software assistant takes the users requirements and goal, and delivers to the user the outcomes that they are interested in. Intelligent software assistant is one of the emerging technologies. They are being used in handheld devices, web technologies, decision support systems and applied in businesses as well.

CALO (Cognitive Assistant that Learns and Organizes) was an artificial intelligence project by DARPA that attempted to integrate numerous AI technologies into a cognitive assistant. It assists the user with managing documents, contacts, meetings, tasks, scheduling and resource management by observing the user behaviour. It had two major spin offs. One good and famous one is Apple's Siri, an intelligent personal assistant which originated from the CALO project. It uses natural language interface to answer questions, make recommendations, and perform actions by delegating requests to a set of Web services. In business these virtual assistants are used for reservation management, call handling and various other purposes. Second was Trapit, a web scraper that makes intelligent selections of web content based on user preferences.

## 2.5 Intelligent Agents in E-Science

Virtual Assistants are not presently used much in E-Science environment. In this research, we plan to develop virtual research assistants that are designed to autonomously carry out a basic task using an E-Science platform. Once they have been provided with certain goals by the user they will work on their behalf to achieve them and will come up with the results later. User can then modify the assigned task or request if needed and let the virtual assistants do their task. It will improve the research time greatly and by automating it, researcher won't have to select all the data manually.

**MVP** (Multi-mission VICAR Planner) [13][14] is a similar system created by Jet Propulsion Laboratory and used by NASA to automate complex image processing steps on their large set of space imagery. MVP aimed at reducing time by assisting user in creating a model based on simple input. The images firstly need to be processed before further analysis can be carried out. The user inputs the image processing goals, MVP then solves it as AI planning problem by constructing a plan of image processing steps to achieve those goals. The plan, in the form of a VICAR script, is then executed returning the processed image. The user can also modify the VICAR script where necessary. It has significantly reduced the time for expert level image processing. On certain tasks it has reduced from 4 hours to only 15 minutes. MVP can help us to select suitable workflows.

Another work, approached a very time consuming large scale map generalization job by implementing a fully automated workflow. That process was too costly and too time consuming to be done manually [15].

## 2.6 Scientific Workflow Systems

In scientific environments there are procedures which are repeated over and over again. These routine tasks are composed of sub-tasks making an encapsulated unit of work which forms a network of dependencies. Therefore a workflow can be defined as a composite task that includes human and machine sub-tasks which coordinate with each other [16]. A scientific workflow management system is an application which allows to define, manage and execute workflows through the execution of software/sub-tasks whose order is defined by compute representation of workflow logic [17]. Therefore the goal of e-Science workflow systems is to provide a specialized environment for the scientists which simplifies their effort to orchestrate computational science experiments.

There has been a lot of interest in scientific workflows in recent years [18]. A graphical programming environment can be used to compose activities in a workflow, so that the outputs from one stage can be passed as inputs to the next. This forms a pipeline of arbitrary complexity. Specialized programming environment provided by e-Science workflow systems [19] aims at simplification of the programming effort required by researchers to orchestrate a workflow or a scientific experiment. There are many

workflow systems being used in many domains of science. Some of the most popular workflow systems are Taverna Workbench [20], RapidMiner [21], Galaxy [22], Kepler [23] and KNIME [24]. These have most number of workflows on the workflow sharing platform myExperiment.

Bioinformatics is a an interdisciplinary field in which scientists seek to discover useful information from the data gathered in life sciences using computational methods [16]. An example is discovering interesting patterns in the data obtained from results stored in database that can be online or from experiments which are performed in laboratory. This discipline is applying well-known computational tools and new tools to well characterized datasets, in an attempt to improve old methods. Like in many other modern scientific research disciplines, advance and complex analysis is empowered by scientific workflow software. Workflow management systems have been built to ease the execution of workflow tools which use services and data from distributed sources [20], [22], [23], [25]. With the rise of new technologies in life sciences that generate massive amount of data, analysis, storage and retrieval of data is becoming a great technical challenge [26], [27]. Often many bioinformatics tools, which can only be used with web interfaces, are not suitable for the analysis of large scale data sets because they are computationally intensive [28], [29]. Workflow systems have been used largely in life sciences [30]. They are so useful that full-featured workflow systems have been developed to fulfil the demand for workflow management in bioinformatics.

Andruil is designed for data analysis of high-throughput experiments in biomedical research [31]. BioExtract Server is another example of workflow systems for bioinformatics. This is a web-based system for querying biomolecular sequence data. It allows execution of analytic tools on the query results, and construction of workflows composed of such queries and tools [32]. Another system is Galaxy which aims at making computational biology accessible for researchers without programming experience. It was developed initially for research in genomics but now it is used as a general bioinformatics workflow management system [33]. Taverna is although domain independent is now used widely for computational biology and other areas of e-Science.

## 2.7 Other Relevant Research

As our work deals with selecting data from very large size datasets, data mining with multi-agents may also be relevant to some extent. **The MADM (Multi Agent Data Mining)** framework was proposed for use in data mining [34]–[36]. It approaches the data mining problem in a modular way. As the data mining consists of several steps, in MADM various kinds of agents are used for carrying out these steps. For example an evaluator agent to choose best results, a comparison agent to compare various results and a coordinator agent to provide data from various data store to the agents. In this way it divides the data mining task to various agents making it efficient and automatic.

Transparent information integration across distributed and heterogeneous data sources and computational tools is a prime concern for bioinformatics. **Rule Responder HCLS (Health Care Life Science)** [37] provides the end-users with a declarative rule based approach. It facilitates the easy integration of heterogeneous systems as well as provides computation, database access, communication, web services. The rules allow the specifying of where information can be accessed, how it can be processed and how to present it to users. Rules involve the tasks of transforming the general information available from existing data sources into personally relevant information accessible via an e-Science infrastructure. Multi-agents invoke and process these rules and collaborate with each other in the Rule Responder. It can help us integrate data from heterogeneous sources autonomously.

In our research work intelligent agents are needed to select the data that can help us reach our goals. It will require some kind **of filtration and classification.** Information filtering is a technique to identify, in large collections, information that is relevant according to some criteria. There are Multi-agent filtering systems as D-SIFTER and SIFTER-II [38] which uses MAS to filter and classify documents. The performance of these systems is comparable to other information filtration systems.

We may need to cluster the data when selecting the appropriate data to send to suitable workflows. Clustering methods are often computationally very expensive, typically between $O(n^2)$ to $O(n^3)$ where n is the number of documents [39]. There are many multi-agent clustering techniques. Multi Agent Turf System (MATS) for image segmentation which is developed being inspired from animals [40].

## 2.8 Conclusion

It can be seen from the literature review that there is a significant room of improvement in this area as not much work has been done in this domain. Previous approaches have been trying to automate the scientific research process completely. Other approaches tried to minimize time it takes for complex analysis tasks by trying to improve the way workflows execute in a distributed computing environment. We have established that time is the only reason it is becoming impossible for the researchers to analyse the continuously growing amount of data. This thesis intends to put forward a multi-agent agent based approach which aims at assisting the researcher working on large amounts of data.

# Chapter 3 - Model and Methodology

It is seen in the previous chapter that current approaches are not enough to help the researcher work with the increasing quantities of data that are becoming available now. Multi agent systems are helpful in the scenarios where a problem is need to be solved autonomously without involving human by breaking it. Each agent in the multi-agent system takes the part of a problem, solves it, communicate with other agents to solve the actual problem. The problem being addressed in this thesis can be divided into sub problems such as interacting with workflow environment, taking user requirements and data selection. Therefore a way is needed to address this problem using multi-agent systems. This chapter presents the architecture of proposed approach for creating multi-agent based virtual research assistants. It covers the detailed architecture of each agent. The prototype will be implemented in chapter 4 and a test case will be discussed which will be used to evaluate the prototype. This chapter begins with an introduction to the architecture of the proposed approach. The chapter will be ended with presenting overall challenges in implementing the prototype.

Figure 3-1 shows the architecture of the proposed multi-agent system.

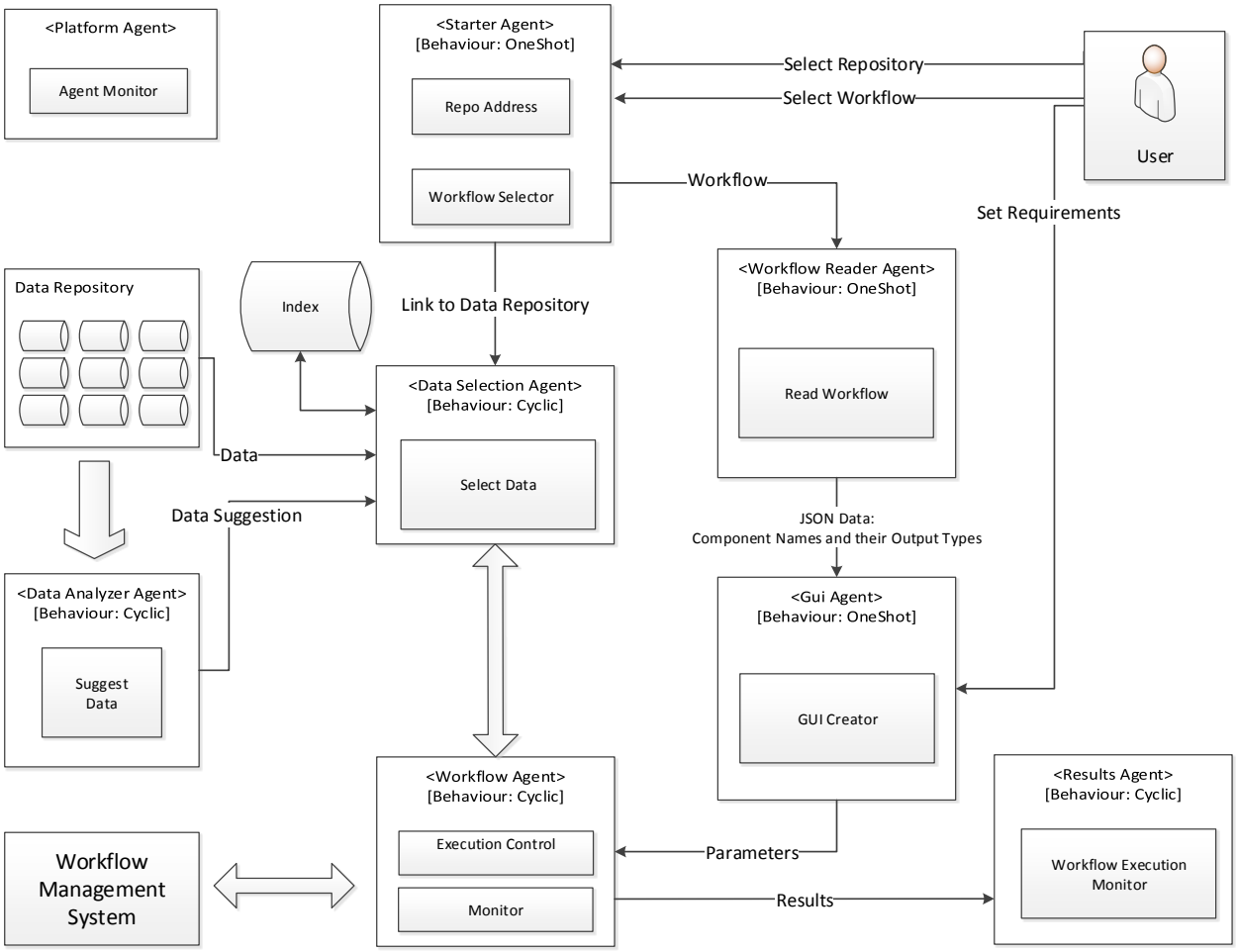


*Figure 3-1 Architecture of the proposed approach*

In this thesis an agent layer architecture is proposed. The idea is to create a Multi-Agent system which stays on top of the underlying workflow management system. This layer will enable controlling and monitoring the underlying system using different agents. In effect the architecture is composed of multiple autonomous agents that perform their tasks independently and communicate and collaborate with each other. Architecture and working of all involved agents is explained one by one.

## 3.1 WORKFLOW AGENT

This agent monitors the execution of underlying workflow system. As a workflow is composed of multiple components, this agent will be monitoring the execution of every component of the workflow. When a workflow is started, components are executed one by one based on their dependencies. They take some input, do any computation using that input, and finally produce an output which is used by next component in the pipeline. Following is the diagram of Workflow Agent.

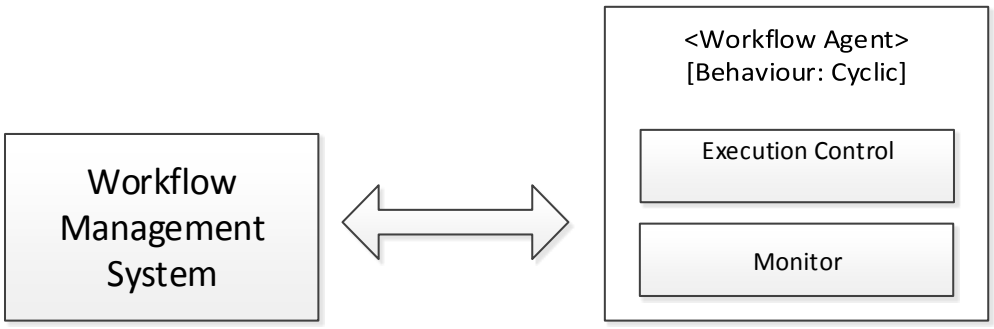


*Figure 3-2 Monitor Agent*

Workflow is responsible for connecting the proposed multi-agent system with underlying workflow management system. It enables the control of the execution of the workflow system as well as monitoring the workflow. The Workflow Management System that is selected needs to enable a way of checking the states of components in response to requests for the monitoring of components. The underlying workflow tool will have an API available which generates events when states of the components are changed e.g. when a component completes its task, in waiting state, throws error or start working. On a change in these states, the agent can request the workflow management system for information about the component whose state has changed. With that state, agent can decide to either change the execution by cancelling it or by giving it new input data to start another execution.

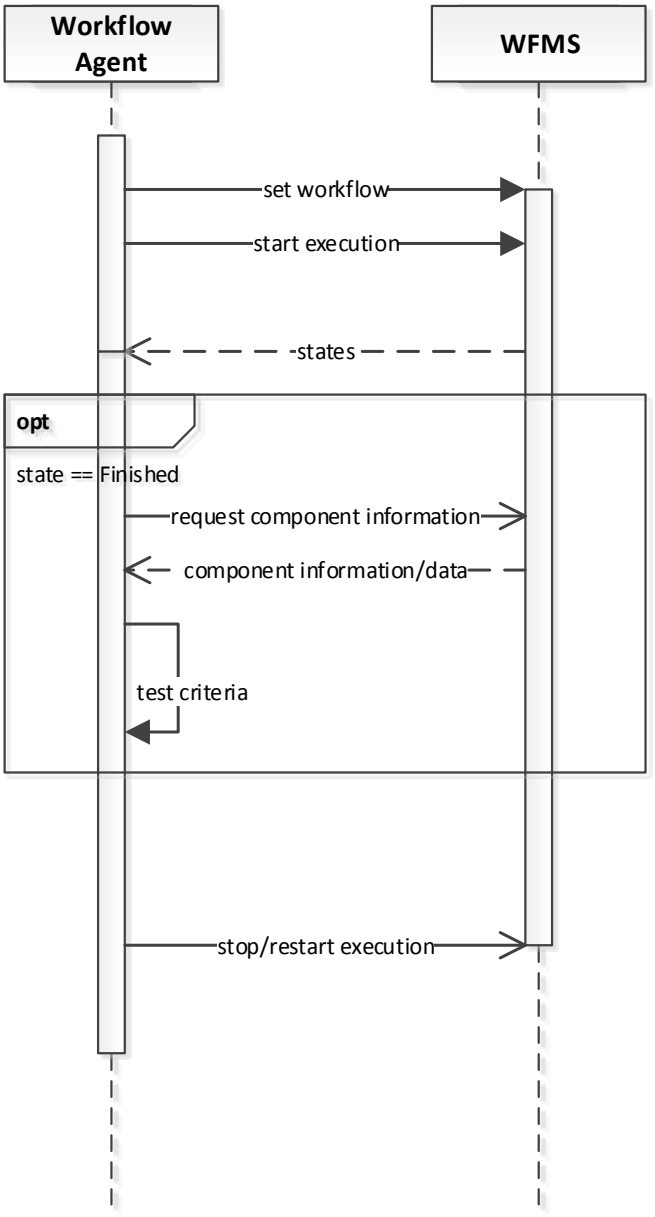


*Figure 3-3 Sequence diagram for Workflow Agent*

This sequence diagram explains working of the Workflow Agent. It starts the execution. When a component starts performing its task, its state is changed and the agent is informed. Looking at the state, agent decides whether to continue the execution as it is, or get more information about that state. In case a component completes its task, agent requests for its outputs. These outputs are then matched

with the criteria defined by user. On a positive match, results will be sent to the results agent from where user can see the desired results.

## 3.2 STARTUP AGENT

When the system is first started, the user will be able to select the workflow he wants to use and the data repository for that workflow. The Startup Agent will bear this responsibility by providing the user with an interface through which to select the workflow and a data repository. The user will be able to select a workflow from the interface which will be then passed on the workflow reader agent. The repository address, will be passed on to the data selection agent which will select the data from there to be given as input to the workflow.


*Figure 3-4 Startup Agent*

Figure 3-5 is the sequence diagram to show the working of Startup Agent.

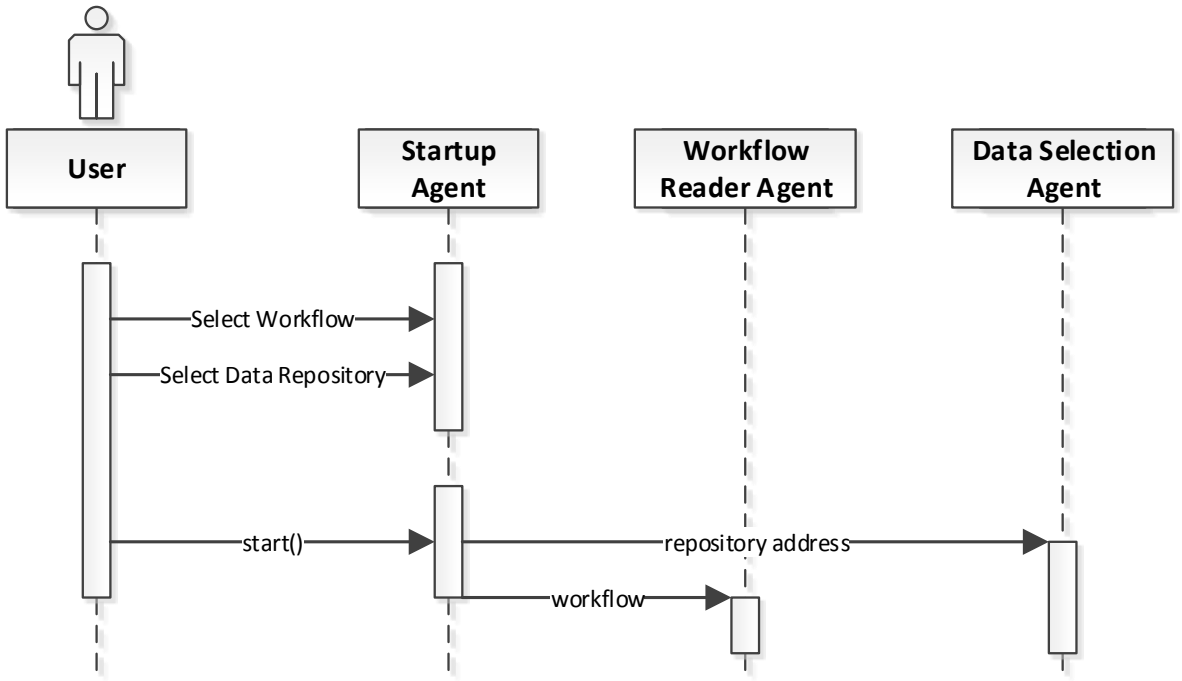


*Figure 3-5 Sequence diagram for Startup Agent*

This sequence diagram explains working of the Workflow Agent. The user starts the system with a given workflow and dataset. After its start-up is complete the agent sends the data repository to the data selection agent and then sends the selected workflow to the workflow reader agent. The Workflow reader agent is responsible for drawing an interface with which user can set the criteria for desired results for various components of the workflow.

## 3.3 Workflow Reader Agent and Settings UI Agent

After a workflow has been selected, it needs to be read by an agent. Workflow definition file contains information for all of its components and how they are connected to each other. Different scientific workflow management systems use different file structure and format to store workflow information. This agent should be able to read the selected workflow format, knowing all the components it is composed of and the inputs and outputs of each component.

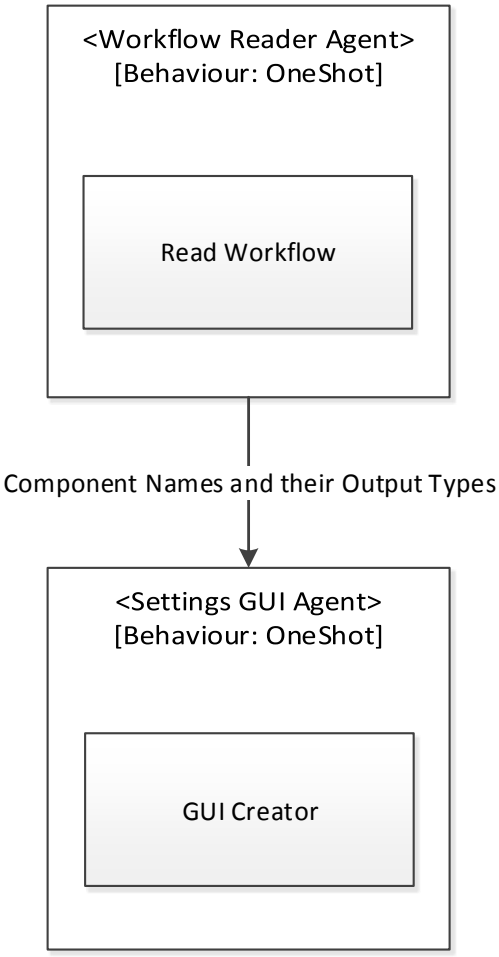


*Figure 3-6 Workflow Reader Agent and Settings GUI Agent*

After reading the workflow, some information is needed to generate a settings interface which is then sent to Settings GUI Agent. This agent only needs to know the components and their outputs to present a UI from where various settings can be selected. Components produce different kinds of output. This agent is able to create appropriate type of user interface for each of them. The user selects all the components whose outputs they want to monitor, and set the required criteria for each item e.g. for string output the user can set define a regular expression. When system is started and workflow is executed, the string outputs from the workflow will be matched with that regular expression which was set using this Settings GUI Agent.

## 3.4 Data Selection Agent

Data Selection is needed if the process is to be automated for the user. The Startup Agent sends a link to the data repository to the Data Selection Agent. It is the responsibility of this agent to select the data from the given data repository and send it to the Workflow Agent from where it will be fed to the workflow. The Data selection agent takes data from the given data repository one by one in different ways. It can pick the data randomly, or in a series or it can select the data using the suggestion from Data Analyzer Agent which analyses the data continuously and suggest next element to be used.

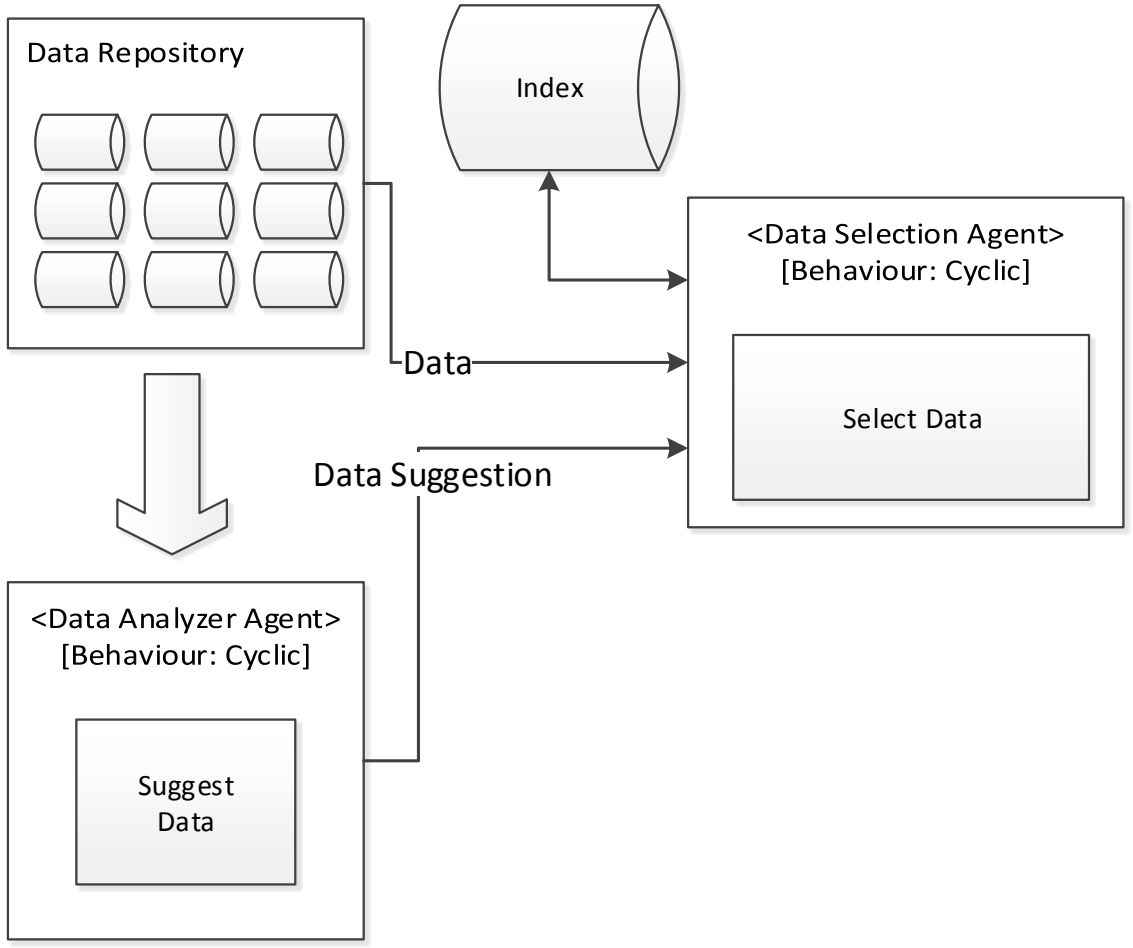


*Figure 3-7 Data Selection Agent*

An Index is maintained for the workflow that is selected and this data will be sent to Workflow Agent as shown in Figure 3-1 which will feed this data to workflow and start/restart its execution.

## 3.5 RESULTS AGENT

When the selected workflow is being executed, its outputs are constantly monitored. An interface is needed which can present the user with the results that they are interested in. The Results Agent does this and displays the results as they start coming from the Workflow Agent. In the end the user will be able to see the desired results from the workflow only instead of being presented all the results which are not useful for him or the results he is not interested in.

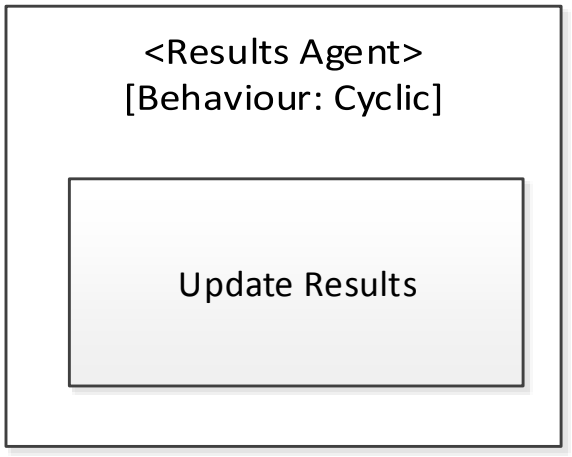


*Figure 3-8 Results Agent*

Sequence diagram in Figure 3-9 shows the working of the Results Agent. From the sequence diagram we can see that while the results are being sent to the Results Agent, the user can opt to skip the current input item on which workflow system is doing its execution by looking at the results. As the results are constantly monitored and user is being notified even with the partial results, being the researcher or analyst he may decide to select/reject the current input sample in processing based on its partial results instead of waiting for more results to come out. At that point, when user choose to skip the current item, workflow agent cancels the current execution and starts processing for another one.

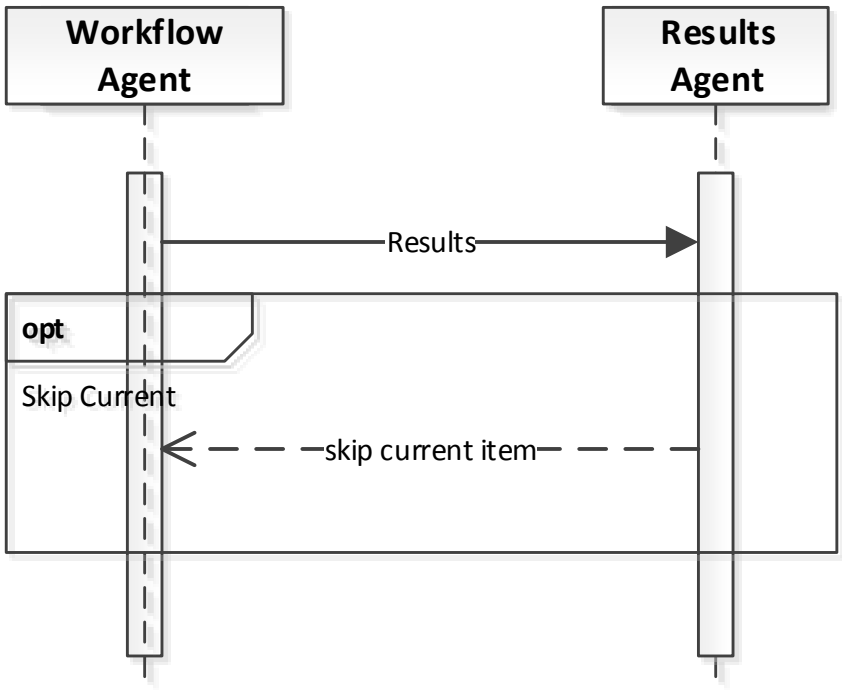


*Figure 3-9 Sequence diagram for Results Agent*

Proposed architecture has been presented and discussed detail in this chapter. Working and behaviour of each agent of the proposed multi-agent approach has been explained. Next chapter will discuss he implementation based on the proposed architecture.

# Chapter 4 - IMPLEMENTATION

This chapter presents the implementation of proposed approach for creating multi-agent based virtual research assistants. It covers the implementation of each module and contribution of other resources used to develop prototype of proposed approach. The prototype will be evaluated in Chapter 5 - and the results will be analysed in order for a conclusion to be reached regarding the validity of the hypothesis. This chapter begins with an introduction to the functionality of the proposed approach. The tools and techniques were used in development of prototypes are discussed next section. The chapter will be ended with presenting overall challenges in implementing the prototype.

## 4.1 IMPLEMENTATION

We needed to choose a workflow system over which we can implement over multi-agent system prototype. It has been seen in literature review that there are many scientific workflow systems developed to perform scientific studies in different domains and for different specific purposes. To prove the proposed approach of the thesis and verify the hypothesis, a simple workflow system was needed. Triana has a simpler user interface than other workflow systems such as Taverna, Kepler or RapidMiner which offer a bulk of features with a cluttered interface. Triana has modular workflow environment which allows components to be developed in Java. Triana worked fine for only some of the sample workflows provided with it. All others created issues making it fail to run properly. The available documentation was insufficiently detailed to give any help on the issue. To get help about what is going on, the developers of Triana were contacted. It was found that all of them have left working on Triana. To date, no updates have been made to Triana.

Taverna 2 and Taverna 1 has the most number of workflows shared on myExperiment [10]. This makes it good choice to implement the prototype on. Another workflow

system, Kepler was also a good candidate for the prototype. For both of them, the available documentation didn't help in programmatically interacting with components of the running workflow while executing. The documentation was not enough therefore people contributing and working on them were contacted using their mailing lists. They were not that responsive to requests for help. It was realised then that in order to develop a prototype meeting the requirements and also have full control over it an in-house workflow system is needed to be built. A workflow system as such would give more control over what can be done with it. To implement the proposed approach anything required by the workflow system could have been developed.

## 4.2 Workflow System

To build a reliable prototype system a relatively simple workflow environment was created. It was built to have more control over interaction of agents with the underlying workflow system. The prototype was built in Java to make it cross-platform and easily compatible with JADE multi-agent framework. Workflow systems usually consists of components/tasks, with each having some input ports and some output ports. A task receives one or more inputs from the input ports it supports, performs its functionality and put the produced outputs in its output ports. Following diagrams show sample workflows from a few workflow systems.

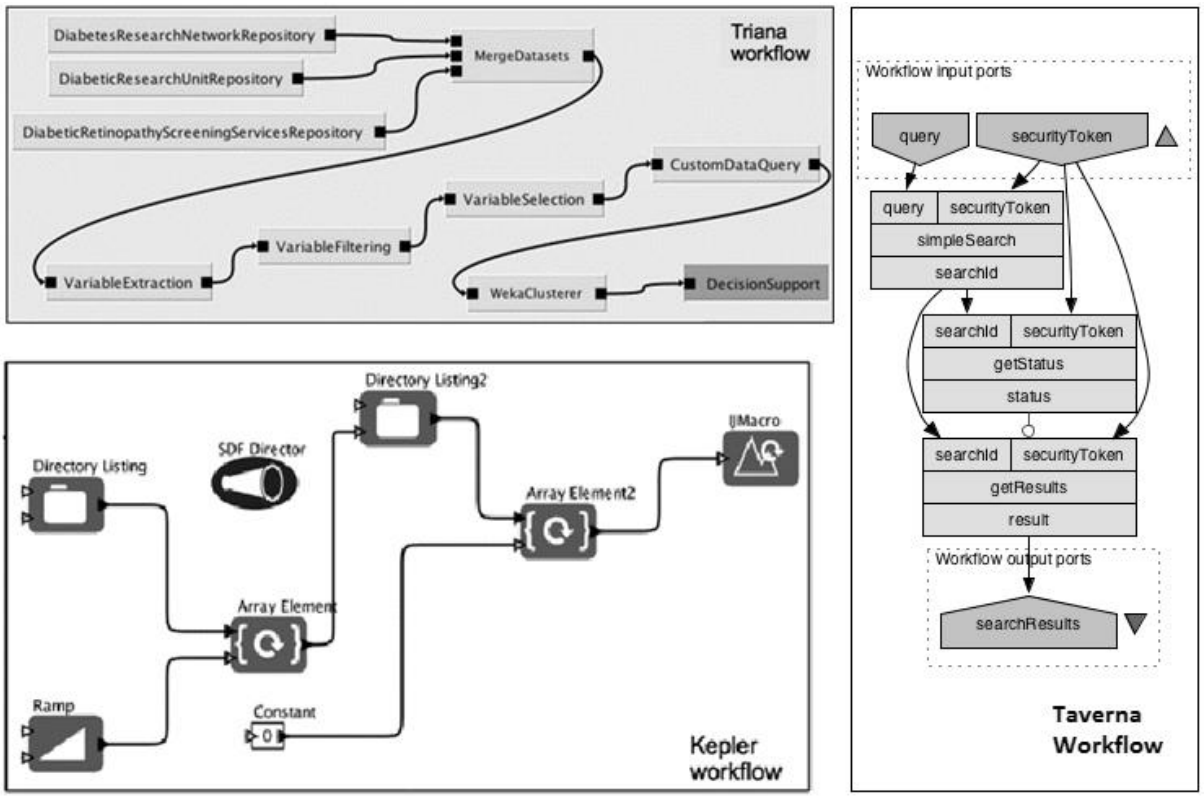


*Figure 4-1 Sample workflows in Triana, Taverna and Kepler*

Each individual task can have its own configurable settings. The developed workflow is composed of three main modules. A main GUI from where tasks can be configured was created. The Workflow Manager is responsible for executing tasks based on their dependencies and Tasks which are basically threads for various components. As the tasks are all threads, they can be executed in parallel. A workflow can be composed in the main GUI using Java code which defines what input each task takes. Any input of a task can be connected to output of another task. The composed workflow is then passed on to the Workflow Manager which is responsible for starting individual tasks and updating the GUI about progress of the individual workflow components. When the program is started all task threads of the current workflow go in waiting state. When execution of workflow is started, the workflow manager initializes all the tasks and resumes execution of head components in the workflow. Any task when it has completed processing, notifies the manager which then decides which task to execute next based on its connections and dependencies. It checks if all parent tasks have completed their execution before proceeding.

## 4.3 Multi-Agent Layer

To implement the multi-agent layer with JADE, a Workflow Agent has been created which connects with the workflow manager module. Each instance of Agent is identified by an AID which is composed of a unique name plus some addresses. AID of the workflow agent is set as “WA”. As the manager knows the state of each component, it notifies WA with each update that is made to the state. The Workflow agent, by looking at the state of the component decides whether to continue or notify any other agent for a particular task as shown in Chapter 3. Agents communicate with each other using ACL message.

To send the data from one agent to another through ACL message, we used JSON. As FIPA specified ACL only allows string, JSON makes it easier to send and receive objects in string format. It is simple and lightweight and has fast parsing. Finally the workflow reader agent which reads the workflow, sends the workflow structure to settings agent as JSON which is then used to generate the settings interface.

## 4.4 Tools and Technologies

The prototype was implemented using diffident technologies. The well designed scenario is chosen after detailed study of design patterns used to develop Multi-agent systems. The detailed architecture has been presented in chapter 3.

Java has been chosen to develop the prototype. It is modern object oriented language based on open public standards. It has an extensive collection of libraries available to be used for free. Also the multi-agent platform JADE, that we have chosen, has been built in Java. C# was another considered choice but there were not much popular and mature workflow systems built with C# moreover it is not open source like Java and most of the tools and libraries written in C# are not available for free. Taverna, the most popular workflow system has been made with Java.

JADE has been selected as a multi-agent framework. It follows FIPA standards and also provides some of its own functionality. Agents in JADE communicate with FIPA compliant ACL. JSON (JavaScript Object Notation) is used to transmit data between some agents where large amount of structured data was needed to be exchanged.

JSON uses an open standard format which is human-readable data objects consisting of attribute-value pairs.

NetBeans IDE was used as the main Java editor. NetBeans supports plugins which help in development. Git with GitHub is used to maintain the code. With GitHub code is always backed up. NetBeans also offers built in Git support and pushing pulling git on GitHub. SmartGit was also used to keep track of commits and code changes.

## 4.5 A Bioinformatics Case Study

Scientific workflow management systems are used in various fields of science to execute a series of computational steps like physics, bioinformatics and astronomy [18]. These systems are widely used to manage computational procedures in bioinformatics projects. Full-featured workflow systems have been developed to fulfil the demand for workflow management in bioinformatics. Scientific workflows managements systems are more applicable in bioinformatics as it deals with analyses of biological data to get information using computer science. In bioinformatics specific analysis pipelines are repeatedly used, particularly in the fields of genetics and genomics.

Workflow systems are used in various domains of science and data analysis as well but they have been so useful in bioinformatics domain that special bioinformatics workflow management systems have been created. Also this research was started with bioinformatics as a potential application area. Therefore a real-world bioinformatics application had to be searched for. To get an example bioinformatics workflow, it was needed to contact the people who are doing it actively.

Firstly a PhD student working in bioinformatics field was contacted. Her research was about PlasmoDB, a biological database which has an advanced search system inherited from EuPathDB. User can build queries with an intuitive user interface like a workflow by defining search steps. This query creation process is known as search strategies. The search query or strategy returns required results for the user.

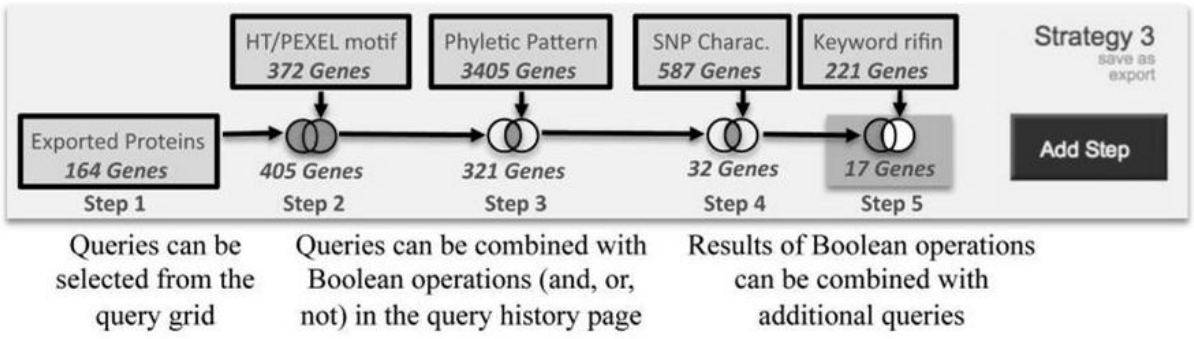


*Figure 4-2 - Creating search strategies in PlasmoDB*

The retrieved dataset can be used for further analyses later. Although PlasmoDB search strategies can be used to perform some processing as part of strategy steps, they are far from a workflow system which allows a lot of different components to do a specific kind of complex processing on the input data. It was found that the researcher was only trying to integrate PlasmoDB and other online databases and not using a workflow on any data.

Although PlasmoDB can be used to perform basic operations on the sequences like translation and finding homology, computationally intensive operations, like protein modelling, cannot performed with it. As this contact could not provide a test case or workflow which could be used for experiments, a test case was still needed. Therefore next the bioinformatics department in ASAB (Atta-Ur-Rahman School of Applied Biosciences) of NUST (National University of Sciences & Technology) was contacted to find some researchers who are doing time consuming analyses on data using the workflow systems. Then that workflow had to be replicated in using the prototype to test how well the system works.

A doctor specialized in Genetics and Genomics/Bioinformatics and System Biology in ASAB was contacted to find any real-world workflow usage example. After several discussions it was discovered that they do analysis that follows a pipeline process but are not using any scientific workflow system. Instead they were wishing for such a system to help executing their routine procedures. After contacting some other researchers working in bioinformatics it came to knowledge that they were not even aware of scientific workflow system but really needed such system to exist. They do use different bioinformatics tools to perform specific analysis operations but where the

whole analyses needed a pipeline process they do it manually. After failed attempts to find some workflow system in use it was realized that any procedure, that can be found, had to be developed manually. Any time consuming data analysis procedure would have been helpful for experimentation using the proposed model. A real-world example was still needed.

A doctor specialized in Molecular Immunobiology was contacted. She provided a simple procedure where they search for sequences on NCBI, get the sequence in FASTA format one by one for each of them and then apply ClustalW for multiple sequence alignment. The workflow was implemented in Taverna as shown in Figure 4-3

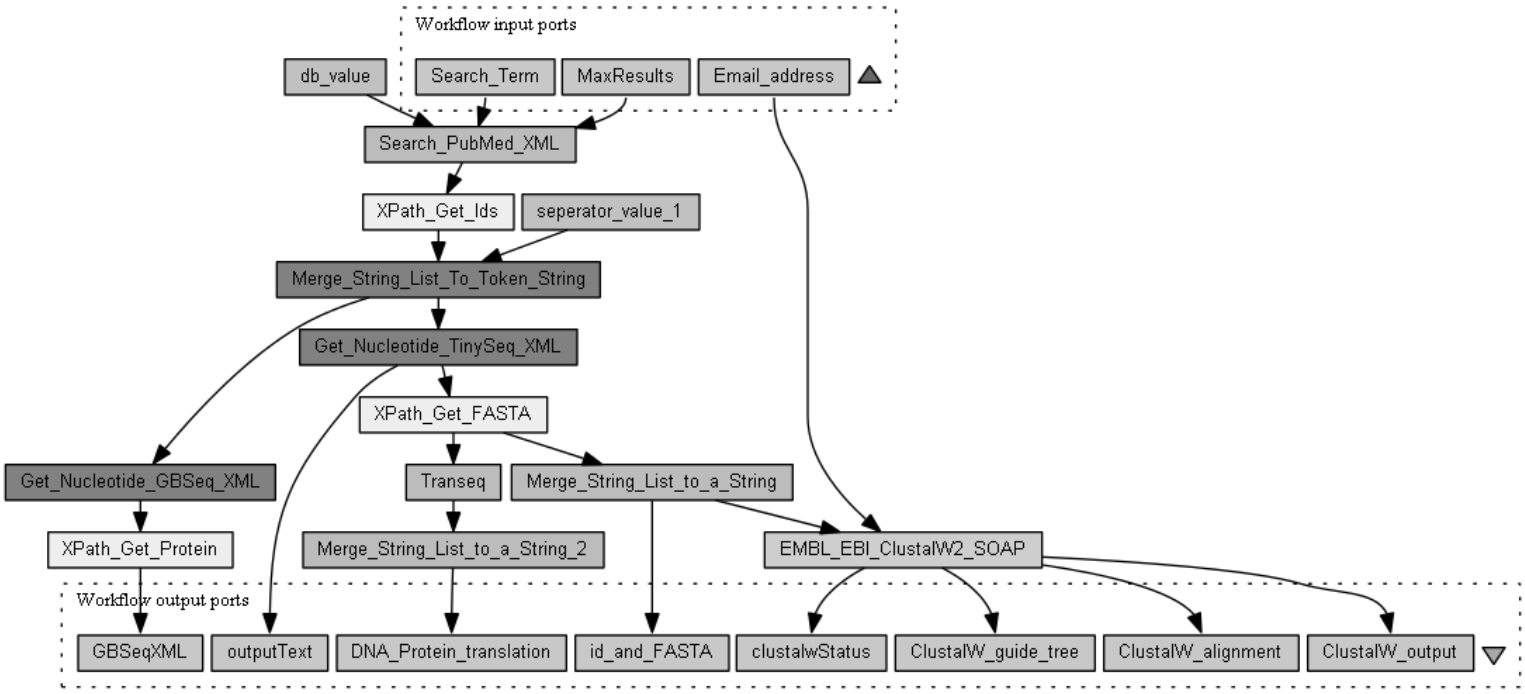


*Figure 4-3 - ClustalW workflow*

## 4.6 Protein Model Prediction

Although this workflow was created but was not enough for our experiments. A better workflow was needed with multiple components and a longer execution time. Multiple components and longer execution time was needed because this research is aimed at helping researchers doing lengthy analysis tasks. We need an example task which is so lengthy and complicated that it is not possible for researcher to wait for completion and repeat it manually again and again until something useful is found. A student researcher was contacted who was doing an analysis on DNA/RNA sequences that takes so much time that it cannot be performed on many sequences. To understand the process and problem multiple meetings with the researchers involved in the process were conducted.

The analysis they were doing on proteins is known as protein model prediction. They do this process to find the functional importance of the protein. In actual analysis a process called crystallography needs to be done. Protein Crystallography determines the atomic structure of protein molecules, in order to reveal the molecular mechanism of highly organized biological systems. In this process crystals of the proteins are created. These crystals are then used to study the molecular structure of the protein. As crystallization is a time consuming process, therefore until really needed, protein model prediction is used instead to find if the model structure. Given protein is searched in online database to find its homology which tells if proteins with similar structure exist. In the context of biology, homology is the existence of shared ancestry between a pair of structures, or genes, in different species. Protein modelling services than predict structure of the input protein and create protein models using different algorithms. The models retrieved are then tested for their quality by using other online available tools. These tools give different scores to each model with which researcher decides which models suits best for further analysis. Next step after this model prediction is known as signalling study. In this thesis the protein model prediction has been selected to create the prototype because this gives a complete real-world case scenario which will be used to test and prove the hypothesis.

It was found that they perform all the tasks with hand without any automation involved. For the analysis, the researcher handpick some data, e.g. protein or DNA/RNA sequences, and submit to online services for processing. That web service accepts

user data, put the job in a queue and inform user with a job id. User have to check with particular web service with job id periodically or alternatively provide the email address to get the notification when job completes. After submission user have to wait for long periods of time to get the results and proceed to next step. Some services can even take 2 – 3 days to finish the job and notify the user.

The details of the various steps involved in their process was used to create a workflow model on the prototype. All the components involved in the workflow were developed. These components submit jobs on their respective servers. After submission of the job an identification is retrieved which is then used by that component to check on regular intervals if the job is finished and results are available. As soon as job is finished, that particular component obtain the results from the server.

Figure 4-4 explains the various steps and components involved in the workflow. This diagram shows the process followed by the researchers. Before going through the process they select a sequence. The sequences obtained could be the result of their research. In this case the sequences were searched for on NCBI database and selected based on the given search query. The search terms returns a number of sequences. For their work they hand pick some sequences and follow the analysis process on them.

After selecting a sequence it is submitted to an online sequence translation service which translates the input to protein. In our case study this was done manually by open the translation service website and copy pasting the sequence. The translation (protein) is then copied back. Then this protein is submitted to NCBI Blastp to find the homology. Based on homology less than or greater than 50% the protein is submitted to different protein modelling services. Different modelling services take different times to complete the modelling jobs. After obtaining the protein models, by manually downloading them from their respective services, they are then submitted one by one to various model verification services. These services return different scores for the submitted models which are then used by the researcher in the end to decide which protein model is useful for their further analysis and processing.

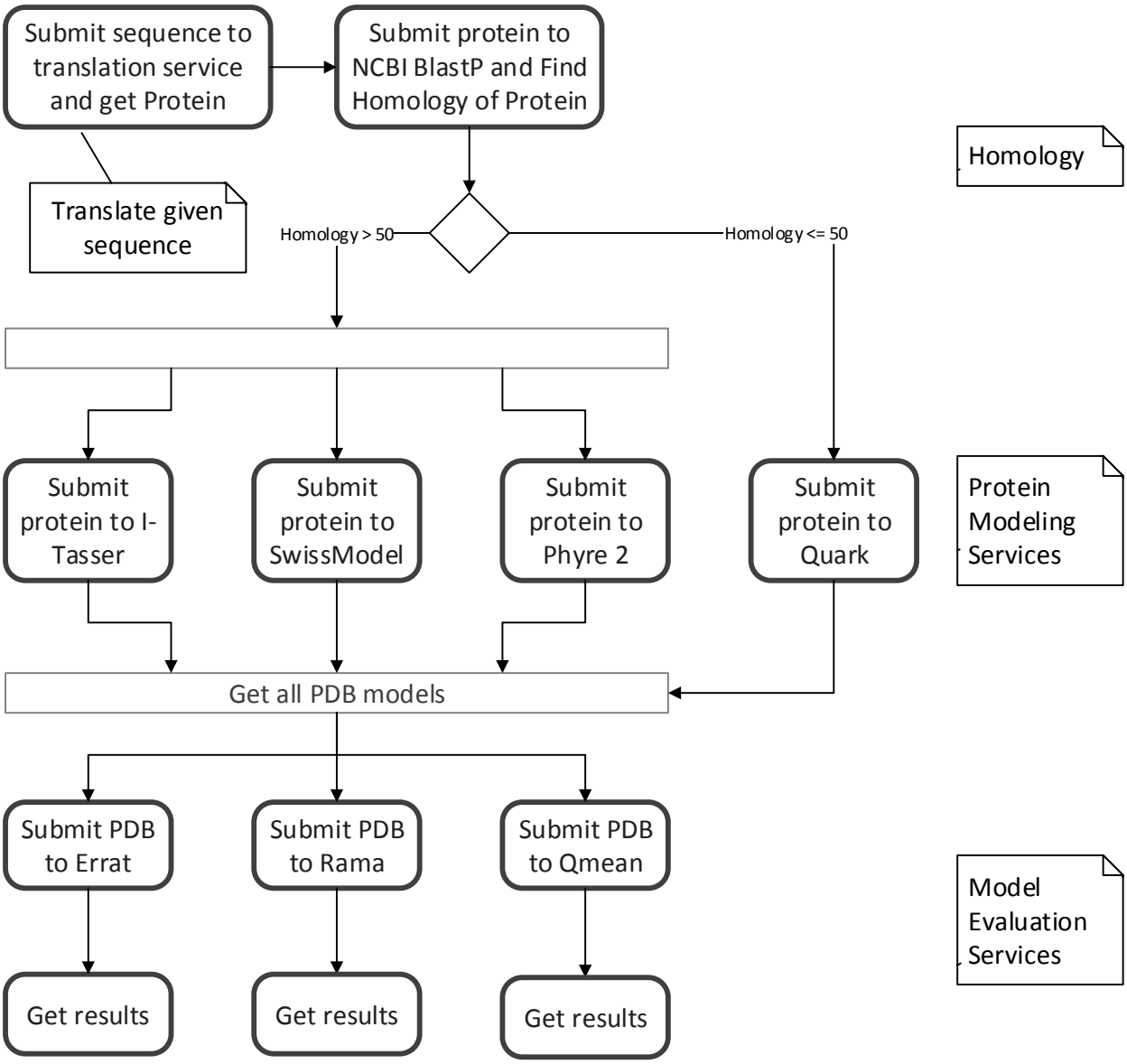


*Figure 4-4 – Workflow of the case study*

It can be seen clearly that being human, a researcher cannot always check these services periodically on regular intervals to get the results when available. A researcher/analysts works hours are lengthy but they still have to take breaks for sleep other routine tasks. A lengthy service may produce the results after the analysts work hour. Being unavailable at a time when the results are produced can add a very significant delay even when notified by email. This analysis task was lengthy enough to be performed on a number of items by the researcher. Also in this case we had a lot of data that needs to be processed to find something really useful this was chosen

as the real-world example. Different setups were created and tests were performed to evaluate the prototype based on this bioinformatics case which will be discussed in Chapter 5.

## 4.7 PROCESS

This sections presents the detail of the each part of process and time it takes to complete that process. In manual processing of the data, researcher/analyst selects the data himself. This selection is usually done by performing some experiments and analyses. The selected sequence is then processed through protein model prediction procedure to get multiple protein models. Those models are then evaluated to select the best of them. Following different parts and components of the workflow are explained in detail.

### 4.7.1 Data Selection

The researcher firstly selects a nucleotide sequence from online nucleotide database (http://www.ncbi.nlm.nih.gov/nuccore) which is a collection of sequences from several sources, including GenBank, RefSeq, TPA and PDB. Genome, gene and transcript sequence data provide the foundation for biomedical research and discovery.

### 4.7.2 Translation to Protein

The selected sequence is then submitted to an online service (http://translate-protein.com/). This website translates the given nucleotide sequence into a protein sequence. This service translates the sequence on client side with JavaScript which makes it complete the process on the press of a button. Researcher copy pastes the sequence and press the button to translate and then copies the resulting protein which takes less than 1 minute.

### 4.7.3 Finding Homology

After translation the sequence, resultant protein is submitted to PSI-BLAST (Protein Specific Iteration - Basic Local Alignment Search Tool) tool provided by NCBI

(http://blast.ncbi.nlm.nih.gov/Blast.cgi) to obtain homologous sequences of three dimensional structures available in protein data bank.

PSI-BLAST by NCBI does not find the homology immediately. Submitted protein is assigned a job id and the researcher has to wait approximately 1 to 5 minutes before it displays the results. In our tests the average time wait times was ~2 minutes.

### 4.7.4 Protein Modelling

Next after finding the homology the researcher submits the protein on different protein modelling services. These services include I-Tasser (http://zhanglab.ccmb.med.umich.edu/I-TASSER/), Swiss-Model (http://swissmodel.expasy.org/interactive), Phyre2 (http://www.sbg.bio.ic.ac.uk/phyre2/index.cgi) or Quark (http://zhanglab.ccmb.med.umich.edu/QUARK/).

Quark is one of the ab-initio modelling techniques which seek to build three-dimensional protein models "from scratch", i.e., based on physical principles rather than (directly) on previously solved structures.

If homology of the given protein was found lesser than 50% than it is submitted to Quark otherwise it is submitted to I-Tasser, Phyre2 and Swiss-Model. Each of these protein modelling services return PDB files of 3D protein models. The PDB (Protein Data Bank) file format is a textual file format describing the three-dimensional structures of molecules.

#### I-Tasser

I-Tasser (Iterative Threading ASSEmbly Refinement) online service (http://zhanglab.ccmb.med.umich.edu/I-TASSER/) is given a protein sequence. It can also be optionally provided email address to notify user by email when the job is done. Other options include password to track jobs by giving password and an optional protein name.

After submitting the protein on I-Tasser it gives the user a Job ID to track the job progress. User can later use this Job ID to check if the job is in the queue, completed or still in progress. Otherwise user can simply give it the email address to be notified when job is finished. When modelling completes it displays users other useful

information about the submitted protein and a list of produced 3d models and also gives user an option to download the models in PDB format. It generates up to 5 PDB models.

I-Tasser saves the models on its server for 3 months. If same protein is submitted again within this time period, it will return the previous results. For any new protein submitted it takes approximately 20 to 60 hours to complete.

#### Phyre2

Phyre2 (Protein Homology/analogY Recognition Engine V 2.0) is another online service (http://www.sbg.bio.ic.ac.uk/phyre2/index.cgi) to model proteins. It requires a protein sequence, an email address and an optional job description. Like I-Tasser on submission of the protein sequence it gives user a job id to track the progress manually. It also notifies the user on email when it is finished. When modelling completes it displays a list of templates it used to build the model and an option to download the final model. It generates only 1 final model. Phyre2 usually finishes within ~30 minutes but can also take up to 2 hours to complete the process.

#### Swiss-Model

Swiss-Model (http://swissmodel.expasy.org/interactive) is another protein modelling server. It requires sequence, an option email address and optional title. It does not take much time to model proteins and completes within ~1-5minutes. It returns 3 PDB models for the given sequence.

#### Quark

Quark (http://zhanglab.ccmb.med.umich.edu/I-TASSER/) is hosted on the same server as I-Tasser. It requires sequence and a mandatory educational email address. Results are sent to the email address and can also be tracked via job id. Optionally it can take name of the protein. Sequence bigger than 200 characters are not accepted on Quark server. On a valid input it takes more than 24 hours to model a sequence and return 10 models when finished.

### 4.7.5 Model Evaluation

The next step after protein modelling is the evaluation of the models. Evaluating a PDB model is done using 3 tools named ERRAT, QMean, ZScore and Ramachandran Plot.

PDB file is submitted on each of these services and they return different values with which the researchers decide which of the PDB model is too be selected.

### ERRAT

ERRAT (http://services.mbi.ucla.edu/ERRAT/) is a program for verifying protein structures determined by crystallography. Error values are plotted as a function of the position of a sliding 9-residue window. Researcher submits the PDB model and in less than 1 minute it displays plots as images including quality factor in a range of 0.00 to 100.

### Rama

Ramachandran Plot (http://mordred.bioc.cam.ac.uk/~rapper/rampage.php) accepts a PDB file and returns a plot for number of residues in favoured, allowed and outlier region. PDB model is selected on the basis of most number of residues in favoured region.

### ZScore QMean

QMEAN Server for Model Quality Estimation (http://swissmodel.expasy.org/qmean/) is used to find QMean and ZScore for the model. It accept PDB model, optional email (because it takes some time to complete) and an optional project name. It takes ~10 – 50 minutes to finish quality estimation. In the end it returns QMean in a range of 0.000 to 1.000 and a ZScore.

## 4.8 Conclusion

This chapter explained how prototype was implemented and detailed the case study used to test the prototype. Case study has been explained in detail with working of different components of the process. Chapter 5 presents the evaluation and results obtained from testing and experimentation on the developed prototype.

# Chapter 5 - Evaluation and Results

## 5.1 Introduction

In the previous chapter the prototype development has been explained. To do experiments and evaluation a real-world scenario was needed on which experiments can be performed to evaluate its usefulness. This would give a practical example of effectiveness of an agent based virtual research assistant in real-world. This chapter will explain how a bioinformatics real-world case was selected for the testing and evaluation of the proposed approach.

The purpose of experiments and evaluation is to judge how well the proposed approach works. The results should validate the performance of proposed approach. Most of research questions would be answered in this chapter on basis of results. The analysis and reasoning on results will be provided in conclusion of this chapter. In contrast of results and their explanation the research question will be answered. Hypothesis will be proven either true or false with the evaluation of results and answers of the research questions. It will be seen in this case study that a workflow managements system was not already being used for the analysis tasks. The example case study has a lengthy analysis task which takes days to complete and it is not practically possible for them to repeat it multiple times. As the developed prototype implements a workflow for their data, it will tested how much time a workflow system can save.

It was stated in the hypothesis that multi-agent based virtual research assistants can be used to automate data selection and workflow analysis. Firstly we need to make sure that multi-agent system can join up and communicate with each other to make up a virtual research assistant. To test this we will run a workflow with our system and

see if Multi-Agents communicate and collaborate together and complete the user's task as expected. In the case study used to perform experiments, no workflow system was being used. Therefore an experiment will be performed first to show how much a workflow system can be useful for manual hand driven complex procedures which involve multiple lengthy steps.

To keep the user from repeating the process again and again for a dataset, it has been proposed in the architecture that MAS will select the data and submit to workflow automatically. Therefore next we will establish how effective can be the automated data selection with multi-agents in a workflow analysis task. To test that an experiment will be executed with the prototype and without prototype system and will be compared to see how much automated selection reduced the time to complete the process. It is needed because it has been assumed that proposed approach should save the user's time.

In the architecture a convenient way has been proposed for the user to provide criteria for desired results and present the partial results. It is needed to be shown that how much earlier the user is informed about the results he has asked for and how that can help. The intermediate outputs generated by components of the workflow are monitored by an agent. User can also opt to terminate the workflow on matching the user defined criteria for the required results. An experiment will be performed where user will define the expected or required results in the prototype. The time when first result is presented to the user and number of terminations on after the time it takes will determine how useful this approach can be in a real-world scenario.

## 5.2 Metrics

Multiple metrics have been selected to evaluate the prototype and its functionality. For quantitative results, accuracy has been selected as first metric. Accuracy of the prototype will ensure that it is working as expected and returning correct outputs for respective inputs. Prototype will be given some input data for which the outputs are already known. Outputs produced will be compared with known outputs to test accuracy.

The aim of this research is to meet the need of analysing continuously growing amount of data in available time. Data analysis is often very lengthy. User assisting agents, which carry out the tasks by reducing the user interaction from the process, should help complete the analysis in lesser time. We have also seen in other research works which take different approaches to reach the goal and try to reduce the time it takes to complete the process. For example MVP aimed at reducing time by assisting user in creating a model based on simple input. Another work, approached a very time consuming large scale map generalization job by implementing a fully automated workflow. That process was too costly and too time consuming to be done manually [15]. In this research we are trying to provide more time to the researcher or data analyst, time will be a major metric to evaluate experiments for quantitative results. Various experiments will be conducted and will be evaluated on the basis of time by comparing the time it takes in different experimental setups.

Qualitative results of the prototype will test the functionality and quality of the prototype. Other qualitative results include usability testing. To test the usability of the system a usability testing technique will be used to evaluate the usability in the hands of its potential users.

## 5.3 Hardware Specifications

The prototype was deployed on a remote virtual machine. Detailed hardware specifications of the remote machine are listed as following.

| | |
|---|---|
| Processor | Intel Xeon ES645 (2.4 GHz) |
| RAM | 4GB DDR3 |
| Operating System | Window 7, 32bit |

*Table 1 - Hardware specification of the remote machine used for long running workflows*

These hardware specifications were enough to run all the test cases because in the experiments the prototype was composed of only the components which use internet services only. No computation or memory intensive tasks were executed on the client side machine. Also we are not evaluating the processing or computation performance.

## 5.4 Prototype Verification

We obtained the initial nucleotide sequence and its final results produced after analysis from the researcher. In this case, the researcher only used two of the protein modelling services I-Tasser and Phyre 2 and evaluated the obtained models with Errat, QMean and Ramachandran Plot.

It was found that there is a difference between results produced using the workflow system and the results gathered with manual procedure. Results from Phyre 2 models were same but I-Tasser models produced different evaluation scores. On inspection we found that I-Tasser software was upgraded between researcher's work and our testing on workflow system which caused the workflow system to produce different results. Models obtained from Phyre 2 resulted the same values as with a workflow system.

Results provided by researcher.

| **PDB Model** | **Evaluation** | | | | | |
|---|---|---|---|---|---|---|
| | Errat | Qmean | ZScore | Rama | | |
| | | | | Favourable | Accepted | Outlier |
| **Phyre 2** | 89.655 | 0.493 | -1.49 | 89.6% | 6.0% | 4.5% |

*Table 2 - Results provided by the researcher*

Results produced using the prototype.

| **PDB Model** | **Evaluation** | | | | | |
|---|---|---|---|---|---|---|
| | Errat | Qmean | ZScore | Rama | | |
| | | | | Favourable | Accepted | Outlier |
| **Phyre 2** | 89.655 | 0.493 | -1.49 | 89.6% | 6.0% | 4.5% |

*Table 3 - Results produced with prototype*

It can be seen from these results that developed prototype has worked as expected. The results produced by different components of the workflow were same as when those tasks were done manually. In the end of it returned the same results as manual work which verifies the accuracy of the prototype.

As we have established in the beginning that time is the only reason it is becoming impossible for the researchers to analyse the continuously growing amount of data. We have seen that other works have tried to reduce the time for the researchers [13]. We will also evaluate the system on the basis of time by measuring how much time it saves for the researcher.

## 5.5 QUALITATIVE RESULTS

The qualitative results are concerned with quality and functionality of the prototype. The prototype was mainly developed to evaluate how well the main time saving purpose of the proposed approach performs. The prototype executes a lengthy workflow task which performs the same task as the researcher. The focus of qualitative results is on behaviour and functioning of proposed approach. It has been assumed that the prototype is running correctly and giving the same outputs as desired. The qualitative results were gotten to ensure that prototype is performing the desired functionality.

**Proposed Approach Qualitative results**

There are several features in the prototype. For getting fair results it is essential to ensure correct working of these features. The purpose of prototype developing using proposed approach is to run the workflow on dataset given by user and present user the useful outputs. Useful outputs are defined by the user before starting execution by setting conditions for various components of the workflow system. The testing of key functions of proposed approach in prototype is listed in table below.

| **Functionality** | **Desired Results** | **Results** | **Remarks** |
|---|---|---|---|
| Launching prototype and giving data repository | Data is retrieved from the NCBI dataset if the search term is given or from the file containing DNA/RNA sequences. | Data has been successfully retrieved from both NCBI and the input file. | PASS |
| Save multiple search terms as multiple data repositories. | The user should be able to save a list of data repositories and given a choice to select from them later. | The prototyped added multiple addresses. They were selected later to be used in next steps. | PASS |
| Communication between agents | The prototype has multiple agents running in collaboration. They should communicate each other to perform correctly. Agent should send data to workflow, monitoring agent should receive the results from workflow agent and settings from the setting agent. | Agents communicated with each other successfully for the given input and execution. | PASS |

| Functionality | Desired Results | Results | Remarks |
|---|---|---|---|
| Defining criteria for required outputs | User can opt to define a settings and criteria for required results from each component. Settings agent should present user and interface where he can set different criteria for different components involved in the workflow | An interface has been generated for the workflow in use where user was able to set criteria for outputs of the components he is interested in | PASS |
| Monitor Components | Each individual components of the workflow will be monitored for its states. An agent in the prototype should be able to monitor each individual component for all states of a component and know when a component produces output. | All components in the workflow were monitored for their outputs. User agent interface keeps the user informed user what each component is doing at the moment. | PASS |
| Show intermediate results | Intermediate results of the working components should be presented. When a component finishes its execution, its results should be displayed to the user which will help user decide to continue or skip the current execution. | Results were displayed to user successfully on an interface. The interface updates with new results as soon as they are available | PASS |
| Skipping execution manually | User should be able to skip the workflow execution for the current item using the interface. Prototype should get the next data item from data agent and start execution on it. | User was successfully able to skip process on current input with the press of a button. | PASS |

| Functionality | Desired Results | Results | Remarks |
|---|---|---|---|
| Terminating execution based on intermediate results | User should be able to skip the current execution based on intermediate results by defining criteria. The execution on current item should be cancelled if the criteria for a component does not match output produced by component | Monitor agent continuously monitored if the outputs of the components match the user defined settings and cancelled the execution when they didn't match. | PASS |
| Caching | With caching enabled, the results should be fetched from cache instead of actual processing or online submission. The outputs of each component produced should be saved in the database. On a re-run if same input is submitted to the component, results should be returned from the cache | When workflow executed, each component took its time and results were saved in the cache database. On a re-run the results were fetched from the cache instantly instead of online submission. | PASS |
| Show required results | User should be able to tell the system which results he is interested in. Only the required results are presented to the user when available. | Agents communicate with each other to by sending the results and matching with the input criteria. User is presented only the results which were defined by him using generated settings panel | PASS |

*Table 4 Test Cases for proposed approach*

These test cases was executed multiple times to ensure accuracy. When unexpected results or failure was encountered, the prototype was fixed to remove the reason for

that failure. Prototype was tested intensively and it was made sure that it passes all required tests.

## 5.6 QUANTITATIVE RESULTS

The quantitative results will evaluate the prototype on the basis of time taken in different experimental setups. One result, for the metric accuracy, has been presented earlier under prototype verification. Results and evaluations will be discussed with graphs when necessary. While in qualitative results we tested desired functionality of the prototype, quantitative results will present measurable results on our defined metrics.

**Experiments**

Experiments and test cases have been designed to test the effectiveness of the system. In all these experiments different test cases were executed to evaluate the system. It was needed to compare the workflow system with manual hand driven process because in the test case that we used no workflow system was being used. Then it will be needed to see how much time the proposed approach saves over conventional workflow process. That will explain the efficiency of the automated workflow executions. Also as in the proposed architecture a convenient way has been designed for the user to provide criteria for desired results and present the partial results, it needs to be shown how such a system can be helpful for user. It was not possible to execute too many tests because testing is very time consuming and we were limited by how much we could do in available time.

### 5.6.1 Manually VS Workflow System

First test that performed on the prototype was to compare the time it takes in manual hand driven procedure and compare with the time it takes using a normal workflow system without any multi-agent setup of the proposed approach involved.

In manual processing of the data, researcher selects the data herself, which is DNA/RNA sequence in this case. This selection is usually done by performing some lab experiments or from literature or some other analyses on the data. Processing a sequence manually to the end to get protein models is a very time consuming task for researcher and it takes about 7 – 8 days to finish the modelling and evaluation of the selected sequence as told by researchers. It was told by the researcher that this analysis usually takes about 7 – 8 days to complete.

To make that sure, in this experiment, we selected three different sequences and processed them both manually and using the workflow system. To keep the experiment fair and unbiased as much time to the process was given as a researcher would do.

This test presents the effort and time it takes to process the data without using any workflow system approach. This experiment is important for this thesis because it will tell whether a workflow system, which automates a complex process, reduces the analysis time for researcher.

In our manual tests on the sequences it took about 5 – 7 days to finish the process which is roughly same as researcher's time. Following table presents the average time taken in different parts of the analysis process.

| *Process Name* | *Average Time for analyst* |
|---|---|
| *Translation* | 1 minute |
| *Finding Homology* | 5-10 minutes |
| *Protein Modelling* | 4 – 5 days |
| *Model evaluation* | 1 – 2 days |
| *Total* | 5 – 7 days |

*Table 5 - Summary of time taken in different parts of analysis by researcher*

In total it takes about 5 – 7 days to process one nucleotide sequence and find the best PDB model.

Same process when repeated with workflow system, using components that we developed, saved a huge amount of time. In our tests a workflow completes its execution in average 25 hours. I-Tasser takes most of the time, it can typically take from 15 hours to more than 50 hours depending on the server load, waiting time and processing time.

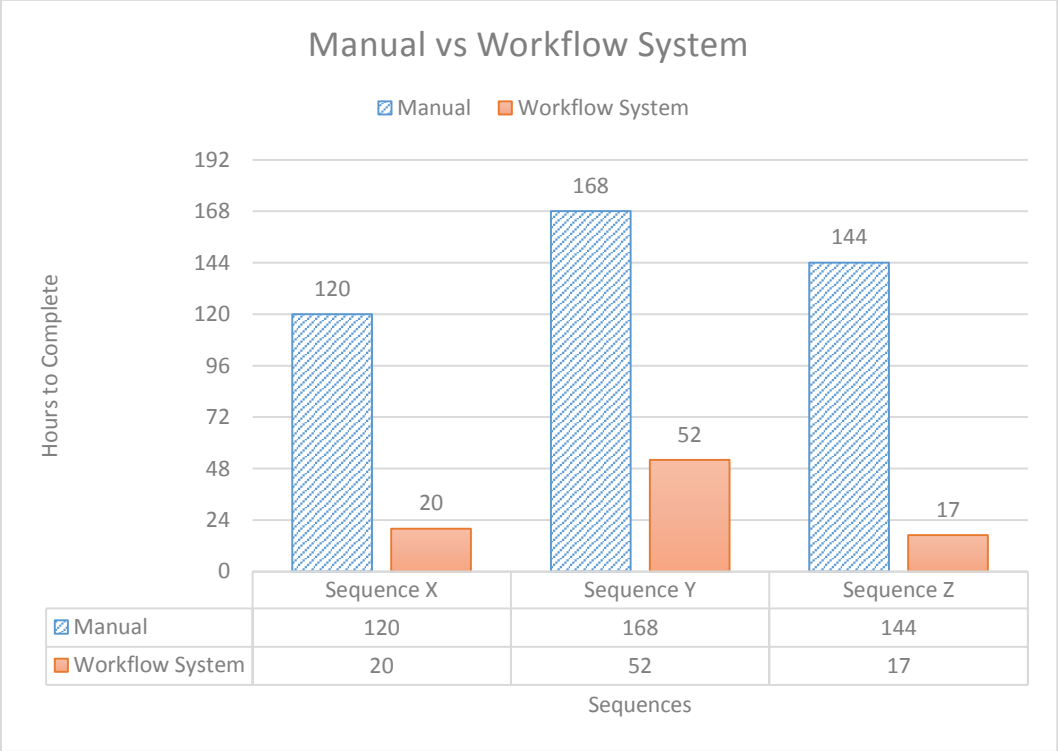


*Figure 5-1 - Time taken in hours to complete the procedure with manual hand driven work VS using workflow system. Time taken for three different sequences is shown.*

In the scenario where all these processes were done manually, the time it takes to get the data from each process depends on the user's availability. Being available at the precise moment at which job finishes is not practically possible for the user. Moreover the researcher cannot always check these services periodically on regular intervals to get the results when available. A researcher/analysts works hours are lengthy but still they have to take breaks for sleep other routine tasks.

Not surprisingly workflow system produces the results in a lot less time than the manual work. Sequence X was processed 6 times faster, Y processed 3 times faster and last sequence 8 times faster. The difference between manual and workflow processed sequences is 100, 116 and 127 hours for sequence X, Y and Z respectively which is equal 4 – 5 days for each. There is a difference of days because when the workflow system was processing the data with one component or another, user was either waiting for one task to complete, doing something else when the task was completed or even unavailable for long time because of sleep or other breaks. This concludes that a workflow system, in our case study, can save 4 – 5 days of analyst time for each sequence to be processed.

### 5.6.2 Manual Workflow VS Workflow with Automated Data Selection

In this experiment we test how much time it will take if user has to run the workflow system multiple times on a bigger data set, which in our case are different DNA/RNA sequences.

This experiment will evaluate proposed approach of this thesis on the basis of time. It will tell how automated data selection without user intervention improves the research time. It will be seen how much time is saved using this approach. Time taken with usual manual workflow execution will be compared with proposed approach where workflow is executed automatically for a given dataset.

In this experiment our proposed approach on workflow system is used to execute a set of sequences. A data repository or a dataset of sequences was needed which can be fed to the prototype. As the researchers select the sequences by searching NCBI nucleotide database we constructed a search term to return a set of sequences from the online database to act as a data repository.

Search term:

```
(Cdkn1b[Gene+Name]+AND+cds[Title])+NOT+partial[Title]
```

This search term returned a set of 16 different sequences which were enough to test the system. Search results can be verified with the following link which searches NCBI Nucleotide database with above search term.

http://www.ncbi.nlm.nih.gov/nuccore/?term=(Cdkn1b[Gene+Name]+AND+cds[Title])+NOT+partial[Title]

In this prototype a user only has to define the search term and maximum number of items which will be forwarded to workflow. These sequences were fed to the workflow system one by one using our prototype automatically.

The following graph and table presents the time it has taken for each execution to complete for each item from the given dataset. The bars display the time it has taken in hours and numbers below represent the sequence number from the dataset.

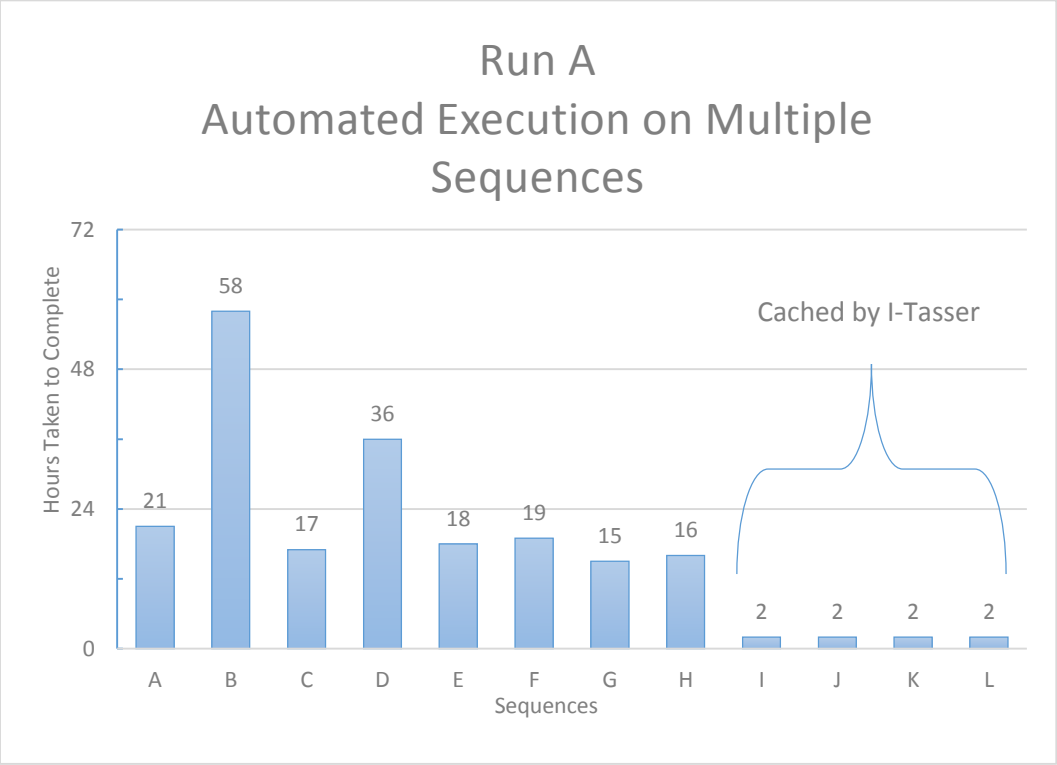


*Figure 5-2 - Time taken in number of hours for different input sequences in a run. Bars represent different executions of workflow on different sequences in that run.*

Four of the input sequences finished in one minute because their homologies were less then defined limit and so they were submitted to Quark which rejected the input for its exceeded length. These finished in less than a minute because no further processing was performed. As they are rejected from the server they are considered failed inputs and are therefore excluded from the chart.

Last four inputs have completed quickly because these inputs have been already processed within previous 8 inputs. Although all initial input sequences were different, some of them have been translated to same proteins. As the proteins are processed in all next steps, I-Tasser returned the PDB models from its own cache instead of modelling them again. These four cached results are shown highlighted in above graph.

In total execution of the 16 iterations of the workflow on a dataset of 16 items finished in approximately 9 days. This process if repeated by feeding data to workflow one by one manually after getting results on each finished execution, will take at least 12 days to complete if researcher is available 12 hours 7 days a week. These 3 more days will be added because any time greater than working hours of the researcher, 10 – 12 hours in our case, will add more hours to make it a full day. Another execution will not be start until researcher gets back to work and do it herself. As two days are off in researcher job, 2 more days will be added making it 14 days.

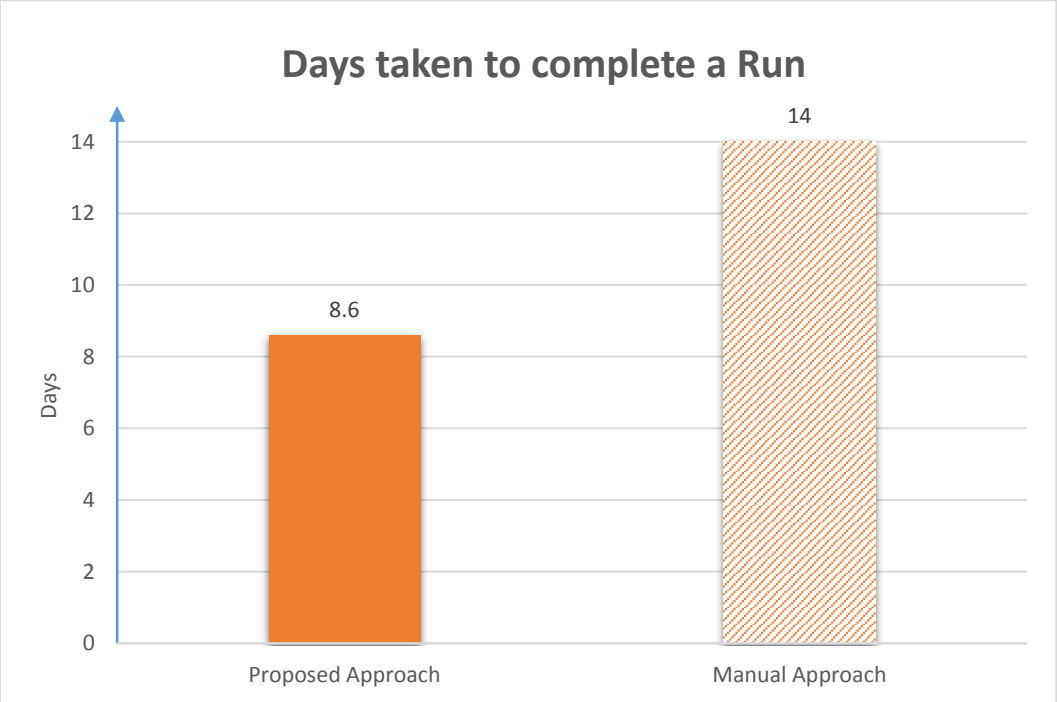


*Figure 5-3 - Comparison of time taken if workflow is executed manually for different inputs VS automated execution with proposed approach*

We can see there is a 48% time difference between these two approaches. Therefore we can conclude that an agent based approach where instead of human being, agents are selecting new data for user and submitting to the workflow application, can reduce analysis time up to approximately 48%.

**Partial Results**

We are trying to reduce the time it takes for researcher to get the results. The prototype is monitoring the outputs produced during the workflow execution. We also monitored how much time it takes between first partial results are shown and last result which finishes completes the process.

When all models are retrieved they are submitted to model verification services. Errat and Ramachandran Plot evaluate the models within 5 seconds on average. QMean server takes time in evaluation. In this test QMean took 12 minutes on average for each input protein model. Partial results for each input protein model of the current workflow input, which contain only Errat and Ramachandran plot, are shown within 30 seconds. In this test first full evaluation result for a protein model, with all Errat, Rama, QMean and ZScore, is shown after 5 minutes. Other models of the sequence returned QMean and ZScore results in average 20 minutes.

### 5.6.3 Execution with caching

In this experiment we enable caching and run the prototype on a dataset which has overlapping data from previously used dataset. Earlier we used a dataset of 16 items. In this experiment we use dataset of 22 items. The experiment will find out how much time can be saved when caching is enabled.

To find the effect of caching, the dataset which overlaps our previously used dataset was used to observe how much faster it finishes. To get an overlapping data set, another search term which contains both previous dataset and some new data is used. To keep the results fair and unbiased, cached results from I-Tasser server are removed for input sequences.

Search Term: `(Cdkn1b[Gene+Name]+AND+cds[Title])`

This search term produced a dataset of 22 different DNA/RNA sequences. Figure 5-4 displays the results from prototype with caching enabled.

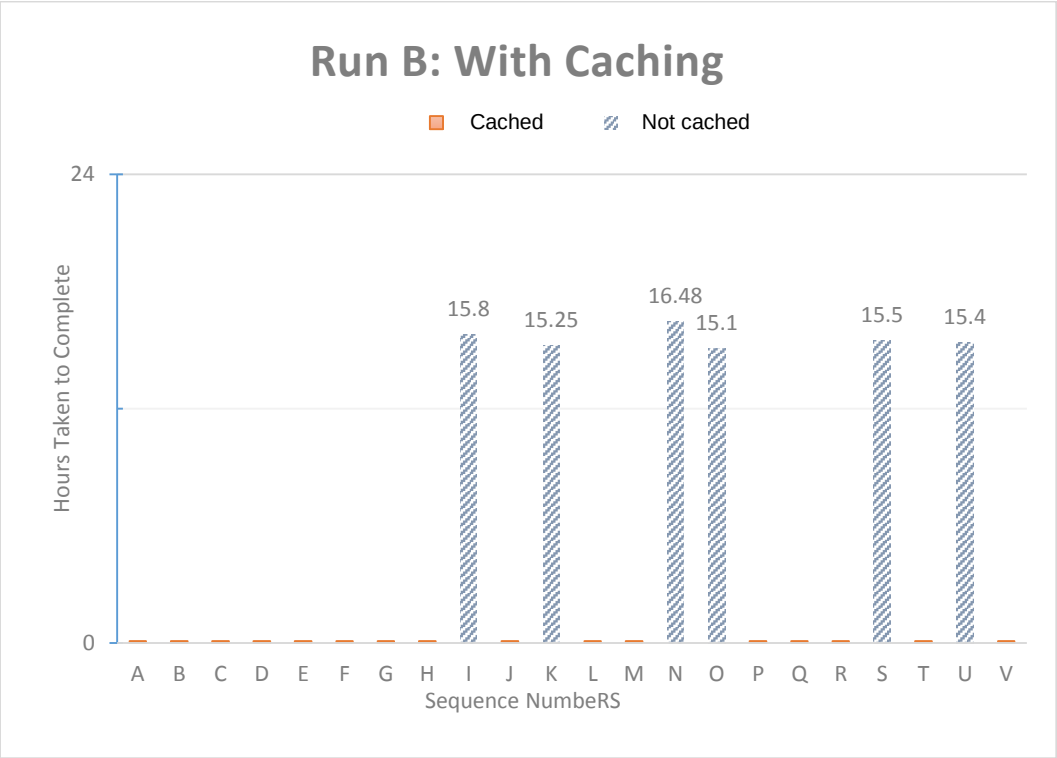


*Figure 5-4 Execution time when caching is enabled. As overlapping data with Run A has been used, only new sequences which were not in cache took full as shown in the graph with patterned bars.*

The chart is displaying time taken for all the inputs. All 22 input sequences were fed to the system and processed one by one. On execution the results are first searched in the cache for the given input and parameters. As every component in the workflow saves its inputs and respective results in a database, items which are found in database were found for all components involved in the workflow and results found were immediately presented to the user. Therefore only non-cached items can be seen taking hours to finish.

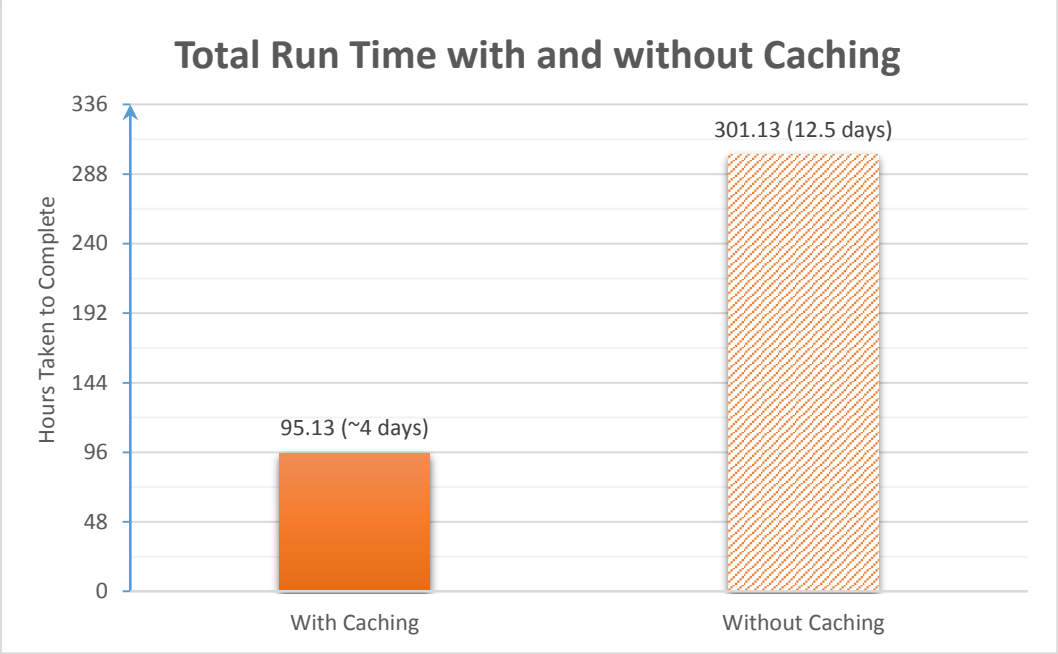


*Figure 5-5 Total hours taken for a run with and without caching*

In above chart, without caching shows the sum of execution time of non-cached items in this experiment and execution time in previous experiment whose results were taken from cache in this experiment. This shows that caching can save huge amount of user time if the executions are repeated in any case.

### 5.6.4 Executing workflow with predefined required results

In this experiment it will be found how much time of the user is saved if user give the agent results desired from the workflow.

In the proposed approach of this research, it was hypothesized that using agent based research assistant can help the user. Prototype displays user an interface based on underlying workflow design, which allows setting criteria for required results from workflow. This experiment is devised to find out how much time it saves and how much helpful such interface is in giving back user the desired results as they become available.

This experiment was devised to find out how much time is saved for the user if he decides to terminate the workflow based on intermediate outputs. On discussion with the involved researchers it was found that they will not be terminating the workflow at

any point. The time when first results appear, while more results are coming, is useful as it helps selecting a useful protein model based on partial results. User decides which model is best after looking at all results. Our workflow is designed in a way that all models are retrieved first and then each model is verified in parallel for different parameters to find which protein model can be useful.

We were given following requirements by the analyst. Any output matching any of the criteria should be presented to user.

- Errat score Greater than 80
- Qmean score Greater than 0.4
- Z score between -2 to 2
- Ramachandran plot favourable region Greater than 80%

We set these in prototype using the settings interface and started the execution. Figure 5-6 illustrates the time when first results are arrived during execution on a timeline.

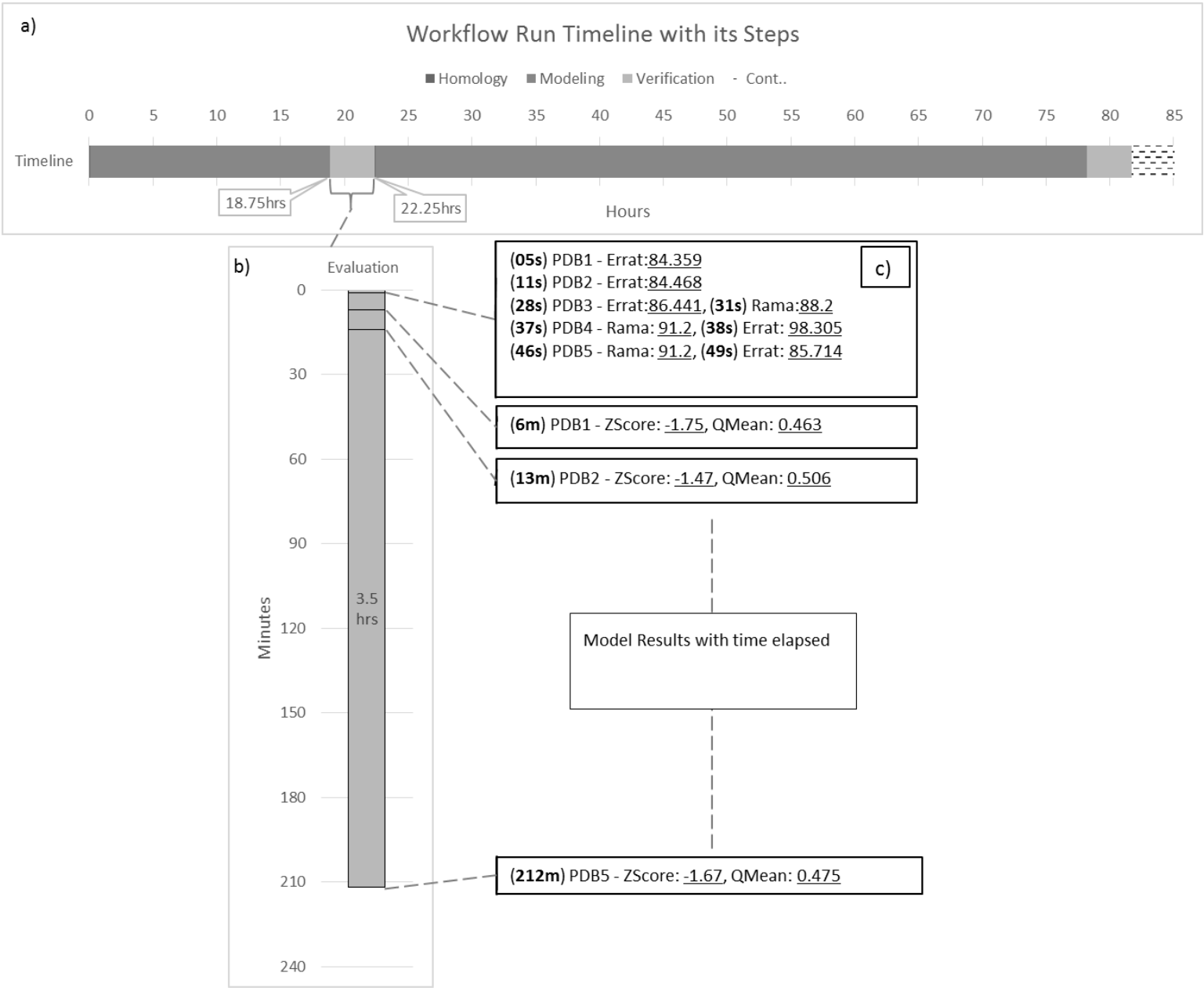


*Figure 5-6 a) Partial timeline of the Run A. Workflow execution is divided in steps as shown in Figure 4-4. b) Evaluation part of workflow was being monitored for matched outputs. Model evaluation took 3.5 hours to finish. c) Timeline of arrival of results is shown as soon as they are shown to user.*

In the firgure Homology and Modeling step takes 18.75 hours and Evaluation step took total 3.5 hours. From the figure we can see that first matching partial result are presented to user in model evaluation step. Errat and Rama evaluated the model a lot faster than QMean server, which provides both QMean and ZScore for the input model. User is presented with the very first matching result in 18.75hrs and 5seconds.

These 5 seconds are counted when model evaluation step starts. This is the step which start returning results. The rest of the Errat and Rama scores for all produced models are presented to user within 1minute. QMeans processes the models at a slower rate. The second matching result is presented to the user after 6 minutes, the third interesting result in 13 minutes. The last result presented for this execution is shown after 212minutes of completion of modelling step.

| Models | ZScore | QMean | Errat | Rama Favourable |
|---|---|---|---|---|
| swiss-nS6XtB-01.pdb | -1.75 | 0.463 | 84.359 | 74 |
| swiss-nS6XtB-02.pdb | -1.47 | 0.506 | 84.468 | 71.2 |
| swiss-nS6XtB-03.pdb | -4.48 | -0.167 | 78.014 | 55.3 |
| itasser-S204880-model1.pdb | -5.82 | 0.248 | 72.007 | 52.8 |
| itasser-S204880-model2.pdb | -7.21 | 0.125 | 73.205 | 57.2 |
| itasser-S204880-model3.pdb | -7.55 | 0.095 | 86.441 | 88.2 |
| itasser-S204880-model4.pdb | -6.69 | 0.171 | 98.305 | 91.2 |
| itasser-S204880-model5.pdb | -7.44 | 0.105 | 57.576 | 76.9 |
| phyre2-db559bd7a7f7acc1-final.casp.pdb | -1.67 | 0.475 | 85.714 | 91.2 |

*Table 6 Detailed table showing all evaluation results for input models. Results matching the user criteria are highlighted.*

It can be seen from

Table 6 that non matching results were not shown to user. It is learnt with this experiment that the way workflow is constructed, upto three and a half user hours (maximum evaluation time in our experiments) can be saved by showing him partial results that match the given criteria, assuming that first evaluation result matches the user input. User can opt to select the model which matches his requirements partially and cancel/skip the execution or wait for until full results are shown. On average model evaluation took one and a half hours in different executions. It is learnt form the results that if workflow was constructed in a way that as soon as a model is produced, it is submitted to model evaluation services instead of waiting for all modeling services to complete, can save a lot more user's time.

## 5.7 Discussion

The aim of this research is to assist user in processing as much data as possible in his available time. Time is therefore the single major factor. Analysts and researchers who need to process a large datasets with lengthy analysis time cannot process all of it. A researcher's time is precious and they cannot work continuously. They need rest, they sleep and take regular breaks. A computer agent instead does not have these limitations. It is always present doing what is requested continuously.

With current approaches user run a scientific workflow and wait for output. He need to repeat the process if results are not good on either new data or with different parameters. Our proposed approach provides user a fairly generic way to automate that. It has been shown with experiments that a multi-agent system can assist user by saving both his time and effort in analysis tasks on large data sets. A researcher can assign agents to do the tasks on his behalf. Instead of waiting for the workflow to complete its execution and proceed on the basis of results, user can let the agents make that decision for him using intermediate results.

It has been shown with the test cases and the results, which improved the way researchers process workflows, that agents do help the researchers perform their analyses with lot lesser efforts. An agent layer, which is controlling the execution and monitoring intermediate outputs of workflow can inform users with the results early even workflow is not fully complete. Instead of feeding the data to the workflow one by one, agent handles the task on the user's behalf and feed data to the workflow when it finishes execution. Agent is also responsible for cancelling the execution altogether if any intermediate output does not match the user defined criteria.

# Chapter 6 - Conclusion and Future Work

Different experiments were performed to evaluate the prototype. The first experiment proved that manual work takes too much time. An automated workflow system can save users significant amounts of time which can be used productively to enhance their research output, thereby leading to new advances in medicine. It was found in the tests that scientific workflow system outperformed human by processing more than 15 proteins in same time it takes for a human researcher to process only one protein manually. Scientific workflow management systems are built to automate lengthy routine tasks. In our case, where a workflow system was not being used, so we created the workflow and it outperformed the manual work.

The second experiment was performed to see the impact of repeating the execution of workflow for different inputs. It was found that proposed approach performs significantly faster that manually repeating the executions. It let users run the executions even when he is not available. So in essence an agent is running the workflow on his behalf even when he is sleeping or having off days. We stated that agents based virtual assistants can help researcher work with increasing quantities of data. This experiment clearly showed that researcher can let the virtual assistants work on their behalf, and save their time to do other tasks.

This answers the research question RQ3

*RQ3: How can they help researchers in automating tasks?*

Third experiment showed the effect of caching. It was seen that caching can save a huge amount of time by giving back the user previously calculated or obtained results. Instead of repeating a workflow component's operation with previously used inputs, old outputs are presented instead saving huge amount of time. On the cloud where multiple researchers are using the same workflow system, caching can help different users by giving them cached results if component is used with same inputs and input parameters.

Finally it has been shown that with user provided criteria, agents will only show user the results which are useful. Giving back everything wastes researcher's time to find what is useful or what is necessary. Virtual assistant will take user defined criteria, monitor the workflow and will present the user only good results. Instead of running the workflow repeatedly and recording the results on each iteration, agent based virtual research assistant will let the user know only when something interesting comes out. This will keep the researcher's interest to only what is useful. In our workflow structure, models were first collected and then submitted to evaluations services one by one. It was found that an optimized workflow, where models are submitted for evaluation as they become available will save more time.

The results of all the experiments answers the research question RQ4

*RQ4: How well multi-agent based virtual research assistant work?*

In this thesis we proposed an approach to data analysis with help of Intelligent Software Agents, known as Multi-Agent Systems, meet the increasing quantities of data. We proposed a Multi-Agent based layer for scientific workflow systems to help researchers carry out their tasks. A prototype was developed to test the hypothesis. Different experiments were performed on the prototype. It has been shown that Scientific Workflow Systems saves a huge amount of time. An agent-based research assistant can repeat executions on a set of items on user behalf saving a lot of researcher's time. It can also notify user when some useful result, which user has asked for, becomes available.

*“Multi-Agent based virtual research assistants can be used to automate data selection and workflow analysis on E-Science platforms”*

Results of all experiments show that hypothesis is valid.

## 6.1 FUTURE WORK

It has been proven from the results that the proposed approach does help the researchers. As the user is able to provide the criteria for the desired results, it can be extended to support profiling the user requirements. As the system is used more and more, the profile built from useful results selected by user should be able to make the MAS more intelligent and autonomous. Another possible approach could be to suggest more useful data next by analysing how useful the previous inputs were at different steps of the workflow. It has been found from the results that how early user receives the first results also depends on design of the workflow. More parallel constructed workflows where components perform their task as soon as their dependencies finish should minimize the waiting time. Proposed approach was evaluated developing a workflow environment developed to have more control over how agents can interact with the workflow system. This was enough so far but it should also be evaluated using some widely used workflow systems like Taverna with complex workflow constructs and behaviours. Integration with existing workflow tools will be a great help for the researchers.

# REFERENCES

[1] B. Feldman, E. M. Martin, and T. Skotnes, “Big Data in Healthcare Hype and Hope,” *October 2012. Dr. Bonnie*, vol. 360, 2012.

[2] M. Schmidt and H. Lipson, “Distilling free-form natural laws from experimental data,” *Science*, vol. 324, no. 5923, pp. 81–85, 2009.

[3] A. Sparkes, W. Aubrey, E. Byrne, A. Clare, M. N. Khan, M. Liakata, M. Markham, J. Rowland, L. N. Soldatova, K. E. Whelan, and others, “Review Towards Robot Scientists for autonomous scientific discovery,” *Autom Exp*, vol. 2, 2010.

[4] “What is e-science.” [Online]. Available: http://www.escience-grid.org.uk/what-e-science.html. [Accessed: 18-Jun-2015].

[5] T. Hey and A. E. Trefethen, “UK e-Science programme: next generation grid applications,” *International Journal of High Performance Computing Applications*, vol. 18, no. 3, pp. 285–291, 2004.

[6] “neuGrid.” [Online]. Available: http://www.neugrid.eu/pagine/home.php. [Accessed: 18-Jun-2015].

[7] “outGRID.” [Online]. Available: www.outgrid.eu. [Accessed: 20-Jun-2015].

[8] D. E. Rex, J. Q. Ma, and A. W. Toga, “The LONI pipeline processing environment,” *Neuroimage*, vol. 19, no. 3, pp. 1033–1048, 2003.

[9] T. Sherif, P. Rioux, M.-E. Rousseau, N. Kassis, N. Beck, R. Adalat, S. Das, T. Glatard, and A. C. Evans, “CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research,” *Frontiers in neuroinformatics*, vol. 8, 2014.

[10] “myExperiment Homepage.” [Online]. Available: http://www.myexperiment.org/. [Accessed: 18-Jun-2015].

[11] D. De Roure, C. Goble, and R. Stevens, “Designing the myexperiment virtual research environment for the social sharing of workflows,” in *e-Science and Grid Computing, IEEE International Conference on*, 2007, pp. 603–610.

[12] A. Sparkes, R. D. King, W. Aubrey, M. Benway, E. Byrne, A. Clare, M. Liakata, M. Markham, K. E. Whelan, M. Young, and others, “An integrated laboratory robotic system for autonomous discovery of gene function,” *Journal of the Association for Laboratory Automation*, vol. 15, no. 1, pp. 33–40, 2010.

[13] S. A. Chien and H. Mortensen, “The Multimission VICAR Planner: Automated Image Processing for Scientific Data Analysis.,” in *IAAI*, 1995, pp. 41–48.

[14] S. A. Chien and H. B. Mortensen, “Automating image processing for scientific data analysis of a large image database,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 854–859, 1996.

[15] J. Stoter, M. Post, V. van Altena, R. Nijhuis, and B. Bruns, “Fully automated generalization of a 1: 50K map from 1: 10K data,” *Cartography and Geographic Information Science*, vol. 41, no. 1, pp. 1–13, 2014.

[16] D. P. Kreil, “From general scientific workflows to specific sequence analysis applications: the study of compositionally biased proteins,” 2002.

[17] D. Hollingsworth and U. Hampshire, “Workflow management coalition the workflow reference model,” *Workflow Management Coalition*, vol. 68, p. 26, 1993.

[18] J. Yu and R. Buyya, “A taxonomy of workflow management systems for grid computing,” *Journal of Grid Computing*, vol. 3, no. 3–4, pp. 171–200, 2005.

[19] D. Gannon, E. Deelman, I. Taylor, and M. Shields, *Workflows in e-Science*. Springer, 2007.

[20] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn, “Taverna: a tool for building and running workflows of services,” *Nucleic acids research*, vol. 34, no. suppl 2, pp. W729–W732, 2006.

[21] O. Ritthoff, R. Klinkenberg, S. Fischer, I. Mierswa, and S. Felske, “Yale: Yet another learning environment,” in *LLWA 01-Tagungsband der GI-Workshop-Woche, Dortmund, Germany*, 2001, pp. 84–92.

[22] J. Goecks, A. Nekrutenko, J. Taylor, and others, “Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences,” *Genome Biol*, vol. 11, no. 8, p. R86, 2010.

[23] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock, “Kepler: an extensible system for design and execution of scientific workflows,” in *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, 2004, pp. 423–424.

[24] M. R. Berthold, N. Cebron, F. Dill, G. Di Fatta, T. R. Gabriel, F. Georg, T. Meinl, P. Ohl, C. Sieb, and B. Wiswedel, “KNIME: The Konstanz information miner,” 2006.

[25] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, and others, “Pegasus: A framework for mapping complex scientific workflows onto distributed systems,” *Scientific Programming*, vol. 13, no. 3, pp. 219–237, 2005.

[26] G. E. Robinson, J. A. Banks, D. K. Padilla, W. W. Burggren, C. S. Cohen, C. F. Delwiche, V. Funk, H. E. Hoekstra, E. D. Jarvis, L. Johnson, and others,

"Empowering 21st century biology," *BioScience*, vol. 60, no. 11, pp. 923–930, 2010.

[27] L. D. Stein, "Towards a cyberinfrastructure for the biological sciences: progress, visions and challenges," *Nature Reviews Genetics*, vol. 9, no. 9, pp. 678–688, 2008.

[28] A. RO and A. RE, "Will computers crash genomics?," *Science*, vol. 5, p. 1190, 2010.

[29] C. Cantacessi, A. R. Jex, R. S. Hall, N. D. Young, B. E. Campbell, A. Joachim, M. J. Nolan, S. Abubucker, P. W. Sternberg, S. Ranganathan, M. Mitreva, and R. B. Gasser, "A practical, bioinformatic workflow system for large data sets generated by next-generation sequencing.," *Nucleic Acids Res.*, vol. 38, no. 17, p. e171, 2010.

[30] A. Tiwari and A. K. Sekhar, "Workflow based framework for life science informatics," *Computational Biology and Chemistry*, vol. 31, no. 5, pp. 305–319, 2007.

[31] "Andruil." [Online]. Available: http://csbl.fimm.fi/anduril/site/. [Accessed: 24-Jun-2015].

[32] "BioExtract." [Online]. Available: https://bioextract.org/query/index.jsp.

[33] "Publicly Accessible Galaxy Servers." [Online]. Available: https://wiki.galaxyproject.org/PublicGalaxyServers. [Accessed: 24-Jun-2015].

[34] H. B. Zghal, S. Fa"\iz, and H. H. B. Ghézala, "A Framework for Data Mining Based Multi-Agent: An Application to Spatial Data.," in *WEC (5)*, 2005, pp. 22–26.

[35] S. Chaimontree, K. Atkinson, and F. Coenen, "Multi-agent based clustering: Towards generic multi-agent data mining," in *Advances in Data Mining. Applications and Theoretical Aspects*, Springer, 2010, pp. 115–127.

[36] V. S. Rao, "Multi agent-based distributed data mining: An overview," *International Journal of Reviews in Computing*, vol. 3, pp. 83–92, 2009.

[37] A. Paschke, "Rule responder HCLS eScience infrastructure," in *Proceedings of the 3rd international Conference on the Pragmatic Web: innovating the interactive Society*, 2008, pp. 59–67.

[38] S. Mukhopadhyay, S. Peng, R. Raje, J. Mostafa, and M. Palakal, "Distributed multi-agent information filtering—A comparative study," *Journal of the American Society for Information Science and Technology*, vol. 56, no. 8, pp. 834–842, 2005.

[39] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," in *Proceedings of the eleventh international conference on*

*Information and knowledge management*, 2002, pp. 515–524.

[40] L. Sun, J. Yan, Y. Chen, and S. Luo, “A new data clustering using multi-agent turf system,” in *IT in Medicine \& Education, 2009. ITIME’09. IEEE International Symposium on*, 2009, vol. 1, pp. 304–307.