

**TOWARDS A FRAMEWORK FOR  
MODELING FISHERIES SYSTEMS  
FOR INCREASED  
ANALYZABILITY**

May 2013

**Ástvaldur Sigurðsson**

Master of Science in Software Engineering





# TOWARDS A FRAMEWORK FOR MODELING FISHERIES SYSTEMS FOR INCREASED ANALYZABILITY

**Ástvaldur Sigurðsson**

Master of Science

Software Engineering

May 2013

School of Computer Science

Reykjavík University

**M.Sc. RESEARCH THESIS**

ISSN 1670-8539





# **Towards a Framework for Modeling Fisheries Systems for Increased Analyzability**

by

Ástvaldur Sigurðsson

Research thesis submitted to the School of Computer Science  
at Reykjavík University in partial fulfillment of  
the requirements for the degree of  
**Master of Science in Software Engineering**

May 2013

Research Thesis Committee:

Marjan Sirjani, Supervisor  
Professor, Reykjavik University

Sigrídur Sigurðardóttir, Co-Supervisor  
Research Group Leader, Matís

Anna Ingólfssdóttir  
Professor, Reykjavik University

Björn Þór Jónsson  
Associate Professor, Reykjavik University

Copyright  
Ástvaldur Sigurðsson  
May 2013

The undersigned hereby certify that they recommend to the School of Computer Science at Reykjavík University for acceptance this research thesis entitled **Towards a Framework for Modeling Fisheries Systems for Increased Analyzability** submitted by **Ástvaldur Sigurðsson** in partial fulfillment of the requirements for the degree of **Master of Science in Software Engineering**.

---

Date

---

Marjan Sirjani, Supervisor  
Professor, Reykjavik University

---

Sigríður Sigurðardóttir, Co-Supervisor  
Research Group Leader, Matís

---

Anna Ingólfssdóttir  
Professor, Reykjavik University

---

Björn Þór Jónsson  
Associate Professor, Reykjavik University

The undersigned hereby grants permission to the Reykjavík University Library to reproduce single copies of this research thesis entitled **Towards a Framework for Modeling Fisheries Systems for Increased Analyzability** and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the research thesis, and except as herein before provided, neither the research thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

---

Date

---

Ástvaldur Sigurðsson  
Master of Science



# **Towards a Framework for Modeling Fisheries Systems for Increased Analyzability**

Ástvaldur Sigurðsson

May 2013

## **Abstract**

In this thesis we propose a framework for modeling and analyzing fisheries system management. The primary goal of this project is helping in decision making by estimating the impact of changes in regulations. Our framework proposes a modeling approach based on Business Process Management. Our model will be run iteratively and output specific files that are then analyzed using data mining techniques for statistical analysis and prediction. We designed our framework based on the lumpfish model in the context of the European research project EcoFishMan.

# **Drög að ramma fyrir líkanagerð til að auka greiningarhæfni fyrir fiskistofna**

Ástvaldur Sigurðsson

Maí 2013

## **Útdráttur**

Í þessari ritgerð komum við með tillögu að hugbúnaðarramma til að smíða líkön af fiskveiðistjórnunarkerfum með það að markmiði að greina þau og styðja við ákvarðanatöku. Það er gert með líkönum sem meta áhrif breytinga á reglum á valdar breytur. Hugbúnaðarramminn okkar leggur til líkanaaðferð sem byggir á viðskiptaferlisstjórnun. Líkanið keyrir á endurteknið á eldri gögnum og framleiðir sérstakar skrár sem eru svo notaðar við tölfræðilegar greiningar og spár. Við hönnuðum hugbúnaðarrammann út frá grá-sleppulíkani sem gert var fyrir evrópska rannsóknarverkefnið EcoFishMan.

*I would like to dedicate this thesis to my parents, without their support through the years  
I would never have reached this stage of my life. There are no words to fully express my  
feelings of gratitude, but I hope this thesis shows them the respect they deserve.  
I love you guys.*



# Acknowledgements

I would also like to thank some people who helped me with my research: Ólafur Waage, Hilmar Finnsson, Sævar Már Atlason, Gunnar Cortes and Þorsteinn Þorvaldsson.

Advania is the company that provided me with WebMethods and for that I would like to give them my thanks.

Finally I would like to give thanks to my supervisor Marjan Sirjani and co-supervisor Sigríður Sigurðardóttir for their guidance, teaching methods and thoughtful feedback.



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	2
1.2 Outline of the Framework . . . . .	4
1.3 Outline of the Report . . . . .	7
<b>2 Proposed Framework: Orange</b>	<b>9</b>
2.1 Creating a Model . . . . .	14
2.2 Advantages of our Framework . . . . .	15
2.3 WebMethods . . . . .	16
<b>3 Lumpfish Model in the Orange Framework</b>	<b>19</b>
3.1 The Lumpfish Model in Stella . . . . .	19
3.1.1 Pros of the Lumpfish Model in Stella . . . . .	22
3.1.2 Cons of the Lumpfish Model in Stella . . . . .	22
3.2 The Lumpfish Model in Orange . . . . .	23
3.2.1 Pros of the Lumpfish Model in Orange . . . . .	25
3.2.2 Cons of the Lumpfish Model in Orange . . . . .	25
3.3 Input Suggestion Model (Pesto) . . . . .	26
<b>4 Analysis of Lumpfish Model using Orange</b>	<b>29</b>
4.1 Consistency of Outputs in Stella and Orange . . . . .	30
4.2 New Questions . . . . .	32
4.2.1 New Questions Justified . . . . .	33
4.3 Statistical Analysis . . . . .	33
4.4 Prediction . . . . .	35
4.4.1 Question 2-1: Fishing Amount Per Trip for Lumpfish and Other Fish . . . . .	37

4.4.2	Question 2-2: Amount of Fishing Trips . . . . .	40
4.4.3	Question 2-3: Amount of fishing trips per year . . . . .	42
4.5	Tools . . . . .	42
4.5.1	Weka . . . . .	43
4.5.2	timeSeriesForecasting . . . . .	43
4.5.3	Pentaho Data Integration (Kettle) . . . . .	44
4.5.4	Excel . . . . .	44
4.6	Summary . . . . .	44
<b>5</b>	<b>Conclusions</b>	<b>47</b>
<b>6</b>	<b>Future Work</b>	<b>49</b>
	<b>Appendix A</b>	<b>55</b>



## List of Figures

2.1	Pipeline mechanism . . . . .	10
2.2	Example of a Java Service . . . . .	11
3.1	A Causal Loop Diagram of the lumpsucker fishery . . . . .	22
3.2	The lumpfish model in the Orange . . . . .	24
3.3	Functions in the lumpfish model . . . . .	24
3.4	Creating a model in Orange . . . . .	25
3.5	The Pesto model . . . . .	26
3.6	Branching for eta and finding its value . . . . .	26
3.7	Mapping eta to document . . . . .	27
4.1	Revenue and Harvest with and without hardcoding . . . . .	31
4.2	NoJobs and Recruit with and without hardcoding . . . . .	31
4.3	PriceRoe and E with and without hardcoding . . . . .	32
4.4	Catch per trip on average . . . . .	34
4.5	Percentage of trips between months . . . . .	34
4.6	Number of boats in Iceland . . . . .	34
4.7	Number of fishing trips each year . . . . .	35
4.8	Trips per boat on average . . . . .	35
4.9	Size of harbours between years in terms of number of trips . . . . .	36
4.10	Machine learning file for the year 2008 . . . . .	37
4.11	Sum of other fish and sum of lumpfish, prediction for 2011 based on 2008-2010 . . . . .	38
4.12	Sum of other fish and lumpfish divided by trips, prediction for 2011 based on 2008-2010 . . . . .	38
4.13	Sum of other fish and lumpfish, prediction for 2013 based on 2011-2012 . . . . .	38
4.14	Sum of other fish and lumpfish divided by trips, prediction for 2013 based on 2011-2012 . . . . .	39

4.15	Sum of other fish and lumpfish with 95% confidence interval, prediction for 2013 based on 2008-2012 . . . . .	39
4.16	Sum of other fish and lumpfish divided by trips with 95% confidence in- terval, prediction for 2013 based on 2008-2012 . . . . .	40
4.17	Trips per month, prediction for 2011 based on 2008-2010 . . . . .	41
4.18	Trips per month, prediction for 2013 based on 2011-2012 . . . . .	41
4.19	Trips per month, prediction for 2013 based on 2008-2012 . . . . .	42
4.20	Trips per year, prediction for 2013 based on 2008-2012 . . . . .	43
4.21	Orange: Model and Analysis . . . . .	45

# Chapter 1

## Introduction

In this thesis we propose a framework for modeling and analyzing fisheries system management. Our framework proposes a modeling approach based on Business Process Management (BPM) [1]. Our model will be run iteratively and output specific files that are then analyzed using data mining techniques for statistical analysis and prediction. We designed our framework based on the lumpfish model of Sigridur Sigurdardottir [2] in the context of EcoFishMan (EFM) research project [3]. The primary goal of this project is helping in decision making by estimating the impact of changes in regulations. For example, this model can help answering the question about the absence of responsibility from industry and stakeholder groups after the legislation change and make it clear whether there is any improvement in the system by performing analysis using data mining.

Our goal is to integrate the lumpfish model to our framework. By doing so we aim to make the model easier to understand, and provide a friendly user environment. Our approach makes it easier to manage and modify the model by allowing online manipulation of the input of the functions that build up the model. This facilities makes our framework suitable to be used for similar problems. Our framework improves efficiency of creating and re-designing models, which provides improved productivity, reliability, quality and reusability. The framework offers various tools for further inspections of the data entered into the model and the results of the model.

We checked that the model created with our framework to be exactly the same as the old lumpfish model. To verify this we used the same inputs for the model and compared the outputs. We also designed new questions and analyzed the data to find the answers. Finally we wanted to be able to cross reference the input and output of the model. There is a possibility that some of the estimated and hardcoded variables used for the inputs of the model are too high or low and we wanted to make sure that they are correct. To do

this we needed to be able to run the model backwards, supply it with outputs and predict the inputs.

Using BPM improves process quality, user satisfaction and time spent on modeling. The BPM contains the models and allows the user to change or modify each process (represents a running model) with a small effort. The framework is equipped with a good built-in web interface, and can be accessed from any browser and requires no special knowledge in computer science to operate. This user interface completes the framework and makes it more usable for any user.

## 1.1 Background

In the year 2012 a new fishing regulation came into effect in Iceland. This legislation focused solely on the lump sucker fishery and made it obligatory for fishermen to land everything they caught in their nets. Before the legislation the common practice had involved cutting the fish belly open on board, removing the roe sack and then discarding the fish since it was not profitable to bring the fish itself to shore. A bio-economic model of the lump sucker fishery was created and simulated for a 25 year period with the aim of assessing the impact of a non-discard policy [2]. Comparison of the two policies was done with two indicators, the profitability margin from the fishery and the number of man-years in the fishery. To simulate this, a system dynamics approach was applied, with a causal loop diagram [4] being developed to describe how variables affect one another, and finally a model was implemented with Stella [5].

This model was a part of the EcoFishMan [3] (EFM) research project and focused solely on fisheries in Iceland. The EFM project aims to develop and contribute to the implementation of a new unified management system in Europe based on increased stakeholder involvement. It is their hope that this system will find acceptance among stakeholders, within governance, industry and consumers, and therefore have a substantial impact of future fisheries policy. They are looking for an ecosystem-sustainable management system under a precautionary framework that will define maximum negative impact, target elimination of discarding of fish and maintain economic and social viability. EFM is a multidisciplinary project, involving scientists and stakeholders in activities relating to biology, stock assessment, technology, economy, sociology and aspects of fisheries management. It does that by engaging structural problems of the Common Fisheries Policy (CFP). The main concentration is on the following problems, the weakness of policy objectives, the short-sighted decision making system and the absence of responsibility from

industry and stakeholder groups. In this new system, the fishermen are given more responsibility for managing and reporting their own activity. With this system the responsibility for detailed allocation and the control of individual quotas and compliance are moved to a lower level. Giving more responsibility ensures that there will be a higher degree of local ownership to the fish and the data, and the transparency of both decision and malfeasance will be easily discoverable.

The focus of this project was primarily on the system of decision making and the division of responsibilities among policy makers, managers, industry and other stakeholders. The design took into account ecological, economic and social factors. This enhanced implementation of the system and improved cooperation and mutual understanding between policy makers and stakeholders. The model was tested with simulations of case studies, where the effects on the factors were reviewed and examined.

The modeling work was carried out by Sigridur Sigurdardottir, a PhD student in industrial engineering at University of Iceland in collaboration with Kristofer Gunnlaugsson, then working as an assistant researcher at the Institute of Economic Studies<sup>1</sup> (IoES). The data for the model was provided by Statistic Iceland<sup>2</sup> (SI), National Association of Small Boat Owners<sup>3</sup> (Nasbo), The Icelandic Marine Research Institute<sup>4</sup> (IMRI) and interviews with fishermen was carried out by Arnar Mar Buason an employee of IoES. The case study was the Icelandic lumpfish fishery. The lumpfish is a small ray-finned fish which can range from 2-50 cm in length and is found in the cold waters of the Artic-, North Atlantic- and North Pacific Ocean. The fish itself holds little value but its roes are used extensively in Scandinavian cuisine as an affordable alternative to caviar produced by sturgeons. The reason this model was created was due to a legislation change in 2010, which was to take effect in 2012. The change made it mandatory to land all harvest and bring to process in land, where previously everything but the roe was discarded at sea. The aim of the model was to assess the impact of this new legislation, by observing the profitability margin and the number of man years in the fishery and see if and how it changed with the new legislation.

<sup>1</sup> [http://hhi.hi.is/en/institute\\_economic\\_studies/](http://hhi.hi.is/en/institute_economic_studies/)

<sup>2</sup> <http://www.statice.is/>

<sup>3</sup> <http://www.smabatar.is/>

<sup>4</sup> <http://www.hafro.is/>

## 1.2 Outline of the Framework

We propose a framework for modeling and analysis of the lumpfish model of the Fisheries System Management which consist of:

- 1 A modeling technique based on Business Process Management [1] using the tool WebMethods [6], plus a web interface for the model such that a user can observe the processes and do some primary analysis.

- 2 Analyzing tools for

- 2.1 statistical analysis using Excel and Weka [7], and

- 2.2 prediction using timeSeriesForecasting [8] addon for Weka, and also more extensive prediction analysis using Pentaho Data Integration.

- 3 Integration of a web service into the model using WebMethods, for the model to interact with Weka through the web service.

- 4 A new model created with WebMethods to predict the inputs of the previous model using its outputs. This is done because some of the inputs for the previous model are estimated and by providing only outputs we can see if the estimations were correct. The results of the estimation calculation are returned within a confidence interval. The functions in the original model are inversed to create this model.

### 1. Modeling:

We chose WebMethods as our modeling tool for the framework since from comparison with other tools it was the fastest and the most powerful tool for Business Process Management we found. Although WebMethods does not have a free software license, for our research we were given a one year free license by a benefactor. Other tools we tried were yEd<sup>5</sup>, a tool for creating BPMN 2.0 files which we used with jBPM<sup>6</sup> an open-source workflow engine written in Java that can execute business processes. Also we tried using ADONIS<sup>7</sup> a Business Process Analysis tool that supports business process management. WebMethods is also programmed in Java, which was one of the reasons we decided to use it. Java is a common programming language with good support.

If we would not have been given a one year free license to WebMethods we would have chosen to use yEd and jBPM. Those tools combined are very similar to WebMethods and are both free of use. yEd creates the models and jBPM runs them. This takes more time

<sup>5</sup> [http://www.yworks.com/en/products\\_yed\\_about.html](http://www.yworks.com/en/products_yed_about.html)

<sup>6</sup> <http://jboss.org/jbpm>

<sup>7</sup> <http://www.adonis-community.com/>

than creating and executing a model in WebMethods. The tools do not offer the possibility to analyze processes to the same extent as WebMethods and we are not sure other tools can be integrated into their models, but they would be sufficient for creating a framework such as the one proposed in this thesis. Being unable to integrate tools into the framework would create a manual step in the framework for the analysis component. The user would have to do the analysis manually and import the results in to the model and tell the model to continue processing. ADONIS on the other hand was not as easy to use and could offer little of what the other tools provided so we decided it should not be used and was not a suitable replacement tool.

The reason we chose to use Business Processes is because they are ideal for creating the model since they are easy to manage and adapt. Business Processes Management attempts to improve processes continuously. It can therefore be described as a optimization process. Business processes also offer application integration which makes them a powerful tool in integration of additional tools to the model.

The web interface is used to view a process while it runs and after it has finished running. WebMethods has a configuration file for each model created and in the configuration file it can store values that are available for the model to use while running. When an activity (steps of the model) such as calculations is started in the model it displays a sign of blue ring with white dots that it is working on the activity, when it finishes it either displays a red cross if it fails or cannot finish or a green √ that shows that it has finished. As soon as an activity is started or if it has finished the user can access it and edit its inputs. The user can re-submit the information from that step and the model continues its execution. This helps in finding errors since the user can see where the process failed and react to that by either changing and re-submitting the inputs or changing that step in the model (if needed). The web interface can be used by more than one user at a time, so many users have access to their instance of the model without interfering with each other.

## 2. Analyzing:

For the analyzing part of our model we had to be able to create new types of files from the inputs and the outputs. This is done because the inputs and the outputs change with each iteration (years in the model) and the inputs are used again in the beginning of each iteration. The reason we created new files was because the original data was not in the form that could be used for the analyzing we wanted to do. In our model we created new steps to take the inputs of the model and the outputs of the calculations (functions) and create the files we wanted. We had to manually add the type of legislation to the files as a comment. The reason we added the legislation to the files was to make it easier dis-

tinguish what data belonged what peroid of years and we knew it could be beneficial for machine learning approaches.

### **2.1 Excel and Weka:**

The files created by the model were sent to Weka to do a preliminary comparison of data between years. From the results of the preliminary comparison we found impact points (apparent changes) in the data and formed questions for further analysis. The results of those questions were used to create charts (graphs) with Excel.

### **2.2 Prediction:**

For the prediction we use the addon manager of Weka and install an addon called time-SeriesForecasting (tSF). The tSF offers time series analyzing and time series forecasting. Time series analysis is the process of using statistical techniques to model and explain time-dependent series of data points. Time-series forecasting is the process of using a model to generate predictions for future events based on known past events. We use the same file as we used for the analysis, however we split the data into different perspectives for periods of years. We look at the data before the legislation separately and predict the adaptation period (2011), we look at the adaptation period and the active legislation (2012) and predict 2013 (which we have not gotten the data for) and finally we use all the data to predict 2013. For all our questions we predict one year ahead.

In the Pentaho Data Integration (PDI) also known as Kettle, we use a graphical user interface called Spoon. It allows the user to design his own transformations for Weka-models and jobs that can be run with the PDI tools.

By adding a plugin step to Spoon it is possible to use the models created with the time-SeriesForecasting addon in Weka and do additional analysis and get charted results. This allows the user also to use the same model for different data, instead of having to create a new Weka-model for each run in Weka. Additionally PDI provides more detailed results than Weka. PDI also accepts different types of input files and transforms them into the ARFF (Attribute-Relation File Format) for the model to use. It can also output the results into different types of files, such as CSV (Comma Separated Values), XML (Extensible Markup Language), TXT (Text) and ARFF. This can be useful for the user, since it saves time in future runs and makes the data usable immediately for charting and other types of analysis that the user might want to do.



**3. Web Service:** With the integration possibilities WebMethods has to offer we were able to integrate a web service<sup>8</sup> for Weka into our framework. That way we were able to use the output file our model created and map it into web service and do the analysis. The web service could not be used for the prediction part of the project. This was due to the fact that it is still in a preliminary phase and does not work with the addons of Weka.

#### **4. Input Suggestion Model**

After creating the first model we created a new model with our framework to predict the inputs of the first model. With the outputs of the first model we predict what its inputs are. The model we created was smaller than the first model, but we were able to reuse the events from the model and

the configuration file. The functions from the previous model had to be inversed before they could be reused. The reason this model is valuable is because the first model has inputs that are predicted and hardcoded before the model runs and there is the possibility that these estimations are either too high or too low. Outputs are provided to the new model and it uses them on the inverse functions and all the calculations and returns the estimated inputs within a confidence interval. This way the inputs of the first model and the outputs of the Input Suggestion Model can be cross-referenced to give a better understanding if the estimated and hardcoded variables were correctly estimated or not.

### **1.3 Outline of the Report**

The remainder of the report contains the following chapters: Chapter 2 contains an overview of the framework; we explain how our framework works and describe the tools used. Chapter 3 contains more detailed information about the fisheries system management model we described here and the modeling languages used (WebMethods and Weka), and then we explain the changes made to the model when it was implemented in our framework. In Chapter 4 we show that the new framework is able to get the same answers as the previous model. We also present new questions with statistical analysis and predictions and describe the tools used for our analysis. Chapter 5 sums up our findings and thoughts about the framework and its process, and describes potential future work.

<sup>8</sup> <https://adams.cms.waikato.ac.nz>



## Chapter 2

# Proposed Framework: Orange

Our framework (Orange) can be used to build a Business Process Model and WebMethods is our tool for building them. Each model consists of activities, events, edges and can have gateways if needed. Activities are the actions of the model, they can have various components, such as Java services, mapping steps, branches, loops, repeaters, sequences and exit steps, although this is not mandatory. The activity can have no components, but then it does nothing in the model but provide a visual confirmation that the activity was used when the user views a previous run of the model. The activity is not required to have inputs or outputs either, but to move the data of the model between activities, gateways and events; they must be specified as inputs and outputs or they are forgotten by the model. The inputs received by the activity are available for mapping into the activities' components. The components then use the inputs for calculations, evaluations, mapping and looping. To start, intermediate and stop the business process we have events, which send and receive signals. The start event of the model creates data for the model; the data can be read from input documents, the model config or user input. There are also events that occur throughout the process; these are used for throwing and catching exceptions. Edges are used to connect events, activities and gateways together. There can be values (boolean constraints) on the edges and therefore edges can be used to guide the path of the process and determine the navigation of the model. Gateways are used to choose the edges to navigate through.

Here below I will start by explaining what a Java Service is and does and explain the pipeline in more detail since they are the two most frequently used components of the model. Afterwards I will explain activities and events and all their possible components in greater detail, and clarify why they are the corner stones of our model and how they

can be used for any model.

**Pipeline:** Our tool (WebMethods) uses a pipeline structure [6] to store, pass and return any variables and data. It can be thought of as a hashtable wrapped up in a structure that allows access to it through a cursor. A value can be created and inserted in the table anytime by the user; it can be dropped or mapped to another value. The pipeline can also contain documents of the same structure as the pipeline itself. It holds the values of a Business Process Model and is created from the start event of the model. The pipeline mechanism is shown in Figure 2.1. In the Business Process Model the pipeline's values are used to map to the inputs of activities, gateways and events of the model. Within these components the inputs can be used for calculations, transformation and other services. When a component has finished its activities it returns the values to the pipeline.

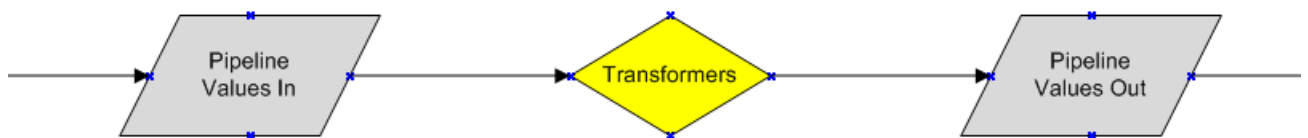


Figure 2.1: Pipeline mechanism

**Java Service:** A Java Service [9] is created by a programmer. It is programmed in Java and can be specified to receive inputs or return outputs. This is not mandatory of the Java Service, as it can run without receiving inputs or returning outputs. The values in the pipeline are connected to the inputs and are used inside the Java Service to create an output. All Java Services have a section for shared code. If a variable or function is declared in the shared section it is accessible by all other Java Services. Figure 2.2 shows an example of a Java Service which accepts a String input and the input in uppercase.

**Event:** For a model to run it must have a start event. The start event can be triggered with several different methods, by a user input in the web interface, by publishing a document to the Integration Server (IS) Broker which listens for changes in document types or by a timed event. The intermediate events do not start or end activities; they happen between activities during the flow of data through the model. They can catch or throw exceptions, they can exist on the boundaries of activities, where they may interrupt or not. The end event stops the flow of data. The end event can throw an error or send a message and will always give a signal based on the event type. The signals are "Failed", "Canceled"

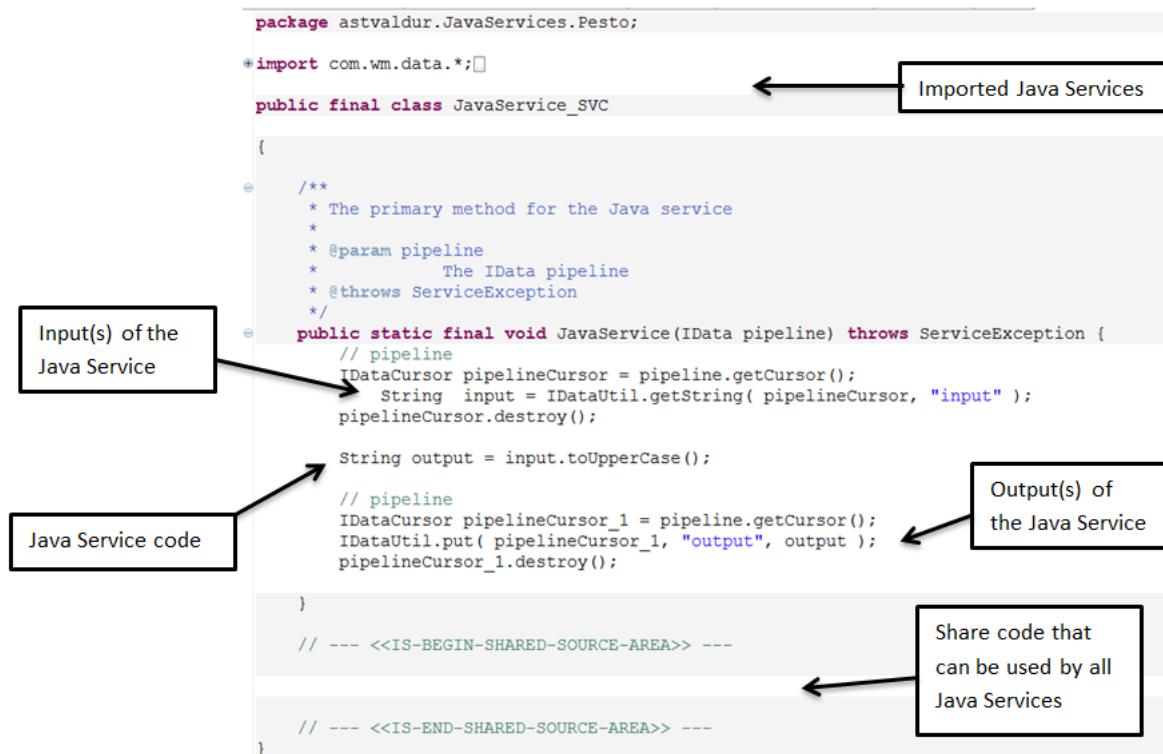


Figure 2.2: Example of a Java Service

or "Completed". After an end event has been processed, no activity can be processed and the model stops.

**Gateway:** Gateways are used to indicate splits and joins in a flow of data and do not have any service associated with them. There are four different types of gateways; each type is designed to work on both sides of the split and join behavior. The standard gateway is an Exclusive (XOR) gateway, the other three are Inclusive (OR), Complex and Parallel (AND). All of the gateways have a main (diverging) behavior which they try to enforce, however if they cannot enforce it they use their secondary (converging) behavior. The XOR gateway diverging behavior is that only one path is taken, the default path is taken if none of the conditions on the other paths are true, however if there is no default path a run-time exception is thrown. Its converging behavior is each inbound path is routed to the outbound path without synchronization just like in the OR gateway. The OR gateways diverging behavior are that all paths that evaluate to true are taken, the default path is taken if none of the conditions on the other paths are true and if there is a run-time exception thrown. The OR gateway merges inbound paths. The Complex gateway diverging behavior is based on transition conditions, but its converging behavior is that it supports join types for AND, OR and XOR gateways. Finally the AND gateway diverging be-

havior is that parallel paths are created without checking conditions, also known as an unconditional split. Its converging behavior is that it waits for all inbound paths.

**Edge:** Edges are used to connect events, activities and gateways together. An edge can also be used to navigate through a model when given a value. If a certain value is true, the model sends information through the edge that holds the value. An edge can also hold a default value, so if no other edge value applies, the default one is taken. Other transition types edges can hold are "Join Timeout", but then it must be connected with a gateway. A join timeout is only available when an activity has multiple input transitions and is only taken when a join timeout period is exceeded. An edge can also have "Step Iterations Exceeded" which is taken when an activity is invoked more times than the indicated limit allows. Finally an edge can have "Unsatisfied Join" which provides a processing path to a downstream activity that can take additional actions when the join experiences a join failure. The edge is not limited to only one type of transition.

**Activity:** Activities are the actions the model performs. They represent the work that is done by the model and are categorized by activity types. Their markers represent specific actions that occur on activity types, and the notation represents the activities themselves using grammar of icons, colors and shapes. The activities are divided into three different activity types. A task activity is referred to as a "task", the subprocess activity as a "subprocess" activity and call activity as a "call activity".

**Subprocess and call activities:** Subprocess and call activities can have loop markers. The loop markers indicate a repetition of an activity, and they can be repeated multiple times. When they are repeated a requirement must be met to stop the looping, but can be set to a maximum number with a loop counter. Subprocess and call activities can have subprocess markers. A subprocess always has a subprocess marker, but a call activity can only have a subprocess marker if it contains a reference to another process which is callable.

**Task:** The task is most used when modeling and it has seven different task types. To add a task to the model the user selects the desired task type from a palette and drags it on to the editor canvas. To change the selected task type the user can right click on it and select a different type. This works the same way to remove a task from the canvas, the user right clicks the activity and selects "Delete". The user has the option to choose an image for each task by right clicking on the task and selecting "Choose Image", this can be restored by selecting "Restore Defaults". Each task can be specified to receive inputs and produce outputs.

The inputs are received from the start event or the previous task in a pipeline format and output in a pipeline format. The pipeline holds all the data for the model, from how long

it took to run to each value at a given time. The first type of task the user can select is an abstract task. It has no run-time implementation, but is rather a task of specialization of activities. A service task is most commonly used for modeling. It is used to call services, from the web or the WebMethods Integration Server. The user task is completed by a human user within the scope of the model, and calls a WebMethods task to provide the user interface for completion. It can present the user with a question that he has to answer or simply require an input to enter. The manual task is not like any other task. It has nothing to process. It has a hand icon which represents the true essence of it, it is done by hand. The user must enter the model, select the step and finish whatever the step requires him to do and signal that the step is ready before it continues.

Rule task can use a Blaze Advisor to create rule in rule tasks in the task, each model is limited to 250 rules. A rule contains a decision table or a rule set with multiple tables. The decision table can contain rule actions regarding data, service and process. Rule actions are of three different types: data actions, service actions, and process actions. A process action is something done to a process and has many options: start, join, suspend, cancel, fail, resume and call a task. These actions are constructed in the rules explorer view and can be dragged from there into the decision tables, which can be used in rules. Finally there are send and receive tasks. They support the Simple Service Protocol (SSP) and are the only SSP support available through the designer. The send task is designed to work with a receive task. The send task delivers a message to a recipient, the receive task.

**Mapping step:** The mapping step is used to create and change values (hardcoding) within a task, map values between different variables and can call a Java Service that are integrated in the WebMethods client, called public Java Services. These public Java services are many and vary from getting the current date, saving the pipeline to a file, connecting to FTP (File Transfer Protocol) or using Java.math operations. A branch is used to examine variables. In each branch there is a switch where the user enters the name of the variables he wishes to examine. If the variables exist in the pipeline, it goes into the branch; otherwise it jumps over the branch.

**Sequence:** Sequences are used in the same way as if-sentences in programming languages. However, unlike the if-sentences, a sequence can be used in two different ways. First of all it can be used with a branch. If a branch is used the sequence can hold a value. This value can be a boolean, integer or numerical value. For example, we could want to know whether a value is true or false and act to according that. A sequence working with a branch can also hold \$default or \$null, if the value is null or no other sequence applies. A sequence can be labeled with the name of the variable, an operator and value *varName* 5 if it is true the sequence is accessed, otherwise it is skipped. Finally a sequence can be

specified to exit in different manors. The values possible to select are: success, failure and done. If success is selected, every component within the sequence must be able to run. If failure is selected it exits with a failure and signals the failure. There is also the possibility to exit on done. Done is mostly used for error handling to make sure every step was taken to log down the error.

**Loop:** The loop is used in a similar way the for-loop is used in programming languages. It can set to time out after a given time, after given iterations or continue looping until it is done. The loop can be specified to receive an input array and return values to an output array. Finally the loop can be iterative as often as a variable value holds (varLoop = 5, loops 5 times).

**Repeater:** Repeaters allow the user to conditionally repeat the sequence of child steps based on success or failure of those steps. They are commonly used to re-execute a set of steps within sequences that fails or to re-execute a set of steps within sequences until they fail.

**Exit step:** The exit step allows the user to specify where to exit from. Some of those options are: exit the entire process flow, a loop, sequence, nearest ancestor, or the parent flow. When the exit step is used it can be specified whether exiting should return a successful or failure condition. If the failure condition is chosen, an exception is thrown. The error message can also be specified by the user or passed into the exit condition from the pipeline.

## 2.1 Creating a Model

All this is available to use with the Software AG Designer and can be accessed from the Service Development (SD) and Process Development (PD) tabs. First the user select the PD and to begin creating a model the user has to start by creating the Java Services he intends to use. Next a empty model is created, and the user starts by deciding if he wants to use a start and end event or a send and receive task and placing them in the model. Then the user creates the activities he needs for his model, note that the user is allowed to use only one activity, but to increase understandability it is recommended to use more steps if the code is large. The user then specifies which input and output each activity should have and connects edges and gateways between them.

If the user is positive that his model is ready, the model can be generated by selecting "Generate". If there are uncertainties in the creation of the model, such as missing edges or steps the PD notifies the user with warnings and errors, but if there is nothing wrong



the activities are created and are accessible from the SD tab. The user selects the SD tab and drags Java Services and other components into the intended steps. The final step is connecting the input and output to the pipeline for each activity and the model is ready to run. If the user wishes to manage his models, he has the possibility to access the config for his model. The config gives the possibility to create, set and overwrite values to be taken additionally into the pipeline of the model. This can be used for the user to control which edges are taken from gateways and which sequences to take.

If the user wants to examine data further, our framework allows the model to output data for different file formats such as CSV (Comma Separated Values), XLS (Microsoft Excel) or ARFF (Attribute-Relation File Format). This can be written in a Java Service to transform the pipeline into the wanted file. Additionally the framework can be connected to data mining and machine learning tools and WebMethods can insert the data into a database, where it can be examined further.

Finally, all the management of the model is available through a web interface that is hosted on a network server and can host more than one user at time. The interface can specify which activities of the model should be logged and how detailed the log should be. Each time the model runs, a process is created which can be viewed by the interface. This process has a unique id that can be searched for, and has the message from the end event so the user can see whether the process has failed, was cancelled or was completed. In the options the user can select whether he wants to be able to edit the process. Editing the process can be done at runtime or after it has finished. It enables the user to edit the pipeline in any selected activity and re-submit from that step. Both results are available after re-submitting and can be compared.

## 2.2 Advantages of our Framework

This proposed Framework for Fisheries System Management (Orange) makes it easy to create new models, whether from old designs or new. It is easy to add, discard or skip activities within models which make it easy to re-use them. The observation of each process is easy and can be used by any user through a very simple web interface which can be used by more than one user at a time. Being able to observe the model at any given time in the process makes it easy to spot errors and find impact points in the data. Being able to edit the pipeline and re-submit from each activity is a powerful way to compare results and helps give insight into importance of variables.

Analysis of data can be done at runtime due to the application integration the framework offers. The framework also offers the possibility to connect to various tools, which that might be beneficial to the model. From any activity there is the possibility with a Java Service to export the data to different file types for further studying or analyzing, for example, using data mining and machine learning. The model can run iteratively, by using outputs of a previous run as inputs for the next run.

The end result of our framework is a simple design, which is easy to look at and understand. It is programmed with Java which is a common programming language, and is available for all platforms and offers good support for whatever the user might need the model to do.

Overall it is our belief that this framework provides the user with an option to create powerful models with various possibilities to both administer and analyze data.

## 2.3 WebMethods

WebMethods is an enterprise software company owned by Software AG<sup>1</sup>, which focuses on application integration, business process integration and B2B partner integration. The WebMethods Integration Server (IS) is one of the core application servers in the WebMethods platform. It is programmed in Java, a common programming language available for all platforms. It offers support such as mapping data between formats and communication between systems. The IS may also be known as the core of the WebMethods Enterprise Service Bus (ESB). The IS supports Java, C, C++ and SQL programming languages for writing services as well as a proprietary graphical language known as flows. The IS also supports a graphical configuration of third party systems operations via the concept of adapter services. It has a web interface that is exposed with HTML for the user to configure and audit solutions and models.

WebMethods has other capabilities to offer as well. In the Protocols/Standards core it offers HTTP/HTTPS (Hypertext Transfer/Hypertext Transfer Secure), FTP/FTPS (File Transfer/File Transfer Secure), and SOAP (Simple Object Access) web services, XML (Extensible Markup Language), LDAP (Lightweight Directory Access) and SMTP (Simple Mail Transfer). In additional packages it is possible to add to it EDI (Electronic Data Interchange)/Flatfiles, XSLT for transforming XML documents, Java Messaging Services, EDIINT (Electronic Data Interchange-Internet Integration), RosettaNet<sup>2</sup>, Mi-

<sup>1</sup> [www.softwareag.com](http://www.softwareag.com)

<sup>2</sup> <http://www.rosettanet.org/>

icrosoft .net<sup>3</sup>, Remedy<sup>4</sup>, SAP (Systems Applications and Products in Data Processing) Enterprise Resource Planning, Siebel CRM<sup>5</sup> (Corporate Responsibility Management) and PeopleSoft Human Resource Management Systems.

WebMethods also offers the WebMethods Business Process Management Suite (BPMS), which unites BPM and service-oriented architecture capabilities to bring an inclusive set of fully integrated tools for automating and managing processes. The BPMS brings together business and IT so they can work together on design, simulations, testing and deploying of processes models. The BPMS uses a single Eclipse-based environment, which allows developers to create and change processes models. The tool can be used by anyone to design a model. The elements of the model can be dragged and dropped and configured when entered into the model. This minimizes the skills need to use the tool and decreases the cost and time of creating a process model. Solutions can be easily stored, shared, administered, analyzed and implemented.

The application integration WebMethods has to offer makes it easy to integrate assets to make more efficient information changes. This works very well with the BPMS and is accessible from the WebMethods Designer.

For these apparent reasons, WebMethods is exactly the tool we were looking for and it serves the demands we made in our goals and will aid us in creating better models.

<sup>3</sup> <http://www.asp.net/>

<sup>4</sup> <http://www.bmc.com/products/remedy-itsm/it-service-management-suite.html>

<sup>5</sup> <http://www.oracle.com/us/products/applications/siebel/overview/index.html>



## Chapter 3

# Lumpfish Model in the Orange Framework

In this chapter we take an example of a model made to observe the lumpfish fishery in Iceland and implement it in the Orange framework. The model was a part of the EcoFishMan research project and the case study focused solely on fisheries in Iceland. It was created due to a legislation change in 2010, which made it mandatory to land all harvest and bring to process in land. There was an interest in studying the impact the legislation change had on fishing. The remainder of the chapter is split up in to three sections; the first section explains the lumpfish model, while the second explains the model in our framework. Finally we introduce a new model called Pesto, that suggest inputs for the lumpfish model.

### 3.1 The Lumpfish Model in Stella

In the lumpfish model, the user specified the inputs of the model with various hardcoded variables, such as a carrying capacity of the boats, fixed cost, biomass growth rate and the tax rate. Some of those hardcoded variables were randomly generated within a certain range that has been predetermined, for example the price of roe and price of lumpfish. The model is based around four functions; The natural biomass growth function, harvest function, cost function and revenue function.

The natural biomass growth is used in the model to account for population dynamics, it is a simple logistic function, as limited biological data can be found on the lumpfish it does not account for age structure. The values used for the function are the stock of

lumpfish, the carrying capacity and intrinsic growth rate of the stock. The growth function is calculated in every iteration.

The output of the harvest function is the amount of lumpfish harvested each year used in the calculations of the two remaining functions. The values used to calculate the harvest of lumpfish roes are ratio of roes in the fish, effort and biomass.

Effort in this case, is measured in number of boats but effort could also be measured in e.g. number of nets or trips. The cost function is an assumption of how much it costs to fish lumpfish. In the model there are two different cost functions, one to calculate how much the cost would be without the new legislation and one with it. The difference between the two functions is that new cost the function for the non-discard policy also takes into account how much it costs to process the catch in land. The values that the functions share are fixed cost, effort, variable cost and resource tax. Finally the revenue function uses the output of the harvest function and is used to calculate the revenue along with the average export price for each kg of roe. The values entered into the function are harvest, utilization coefficient and the price for a kg of roe.

$$(X(t - dt) + (recruit - harvest)) * dt \quad (3.1)$$

$$q * E * X \quad (3.2)$$

$$fc * E + revenue * vc * resource\_tax \quad (3.3)$$

$$harvest * eta * priceroe \quad (3.4)$$

The functions by number: (1) Natural Biomass Function, (2) Harvest Function, (3) Cost Function and (4) Revenue Function [2].

Variables in functions:

- $t$  = Time
- $dt$  = Time Step
- $recruit$  = Biomass Growth Rate

- harvest = Ratio of Roe from Harvested lumpfish
- $q$  = Ratio
- $E$  = Effort
- $X$  = Biomass
- $fc$  = Fixed cost
- $vc$  = Variable Cost
- resource\_tax = Resource Tax
- $\eta$  = Utilization coefficient
- priceroe = Price for each KG of Roe

These functions are the inputs implemented in a simulation tool called Stella [5]. Stella offers a graphical interface for building models and is available for Windows and Macintosh. It is created with Lisp-like language which is strongly typed and object oriented. The software offers an icon-based graphical interface and each entity of the model can be linked to another with arrows and underlying mathematical relationships. The user can also change the input parameters and test different policies. Stella is designed to facilitate symbolic programming tasks in artificial applications. It offers a way to visualize dynamics and communicate on how complex systems work and also explains how entities in the model interact with each other. It has the possibility to display all variables graphically and see how the change over a full iteration. Stella can define sliders to make sensitivity analysis easier. Statistical distributions can be assigned to all relevant aspects of the model. Figure 3.1 shows the causal loop diagram (CLD) [2] used as a skeleton to build the lumpfish model in Stella.

The CLD [4] is a causal diagram that aids in visualizing how different variables in a system are interrelated. It consists of a set of nodes and edges, where the nodes represent variables and the edges the link between the variables. A link marked positive indicates a positive relation and a link marked negative indicates a negative relation. A positive causal link means the two nodes change in the same direction, while a negative causal link means the two nodes change in the opposite directions.

Closed cycles in the diagram are very important features of the CLDs. A closed cycle is either defined as a reinforcing or balancing loop. A reinforcing loop is a cycle in which the effect of a variation in any variable propagates through the loop and returns to the variable maximizing the initial deviation, while a balancing loop returns to the variable a deviation opposite to the initial one.

If a variable varies in a reinforcing loop the effect of the change reinforces the initial variation. The effect of the variation will then create another reinforcing effect. Without breaking the loop the system will be caught in a vicious cycles of circular chain reactions. For this reason, closed loops are critical features in the CLDs.

All additional analysis of the model was done with Excel to find the variables to use as inputs.

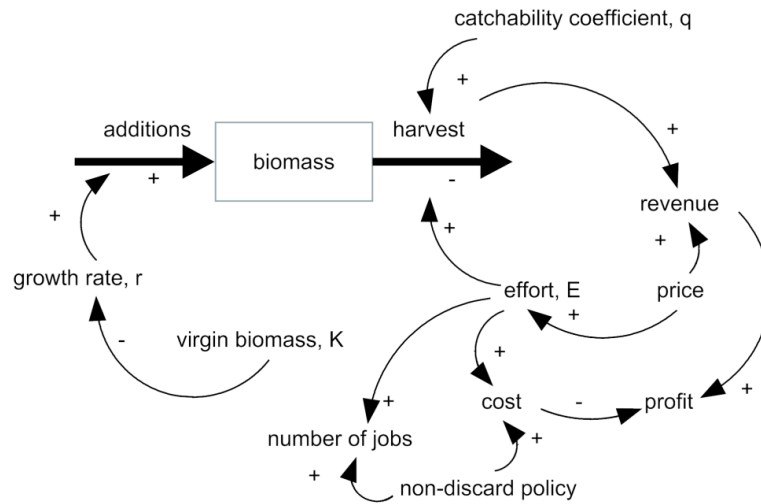


Figure 3.1: A Causal Loop Diagram of the lumpsucker fishery [2]

### 3.1.1 Pros of the Lumpfish Model in Stella

The greatest strength of this model is that Stella is capable to provide information after each iteration and can easily forecast as many years as needed. There is no programming experience needed to create a model in Stella. We consider the possibility to add a graphical interface for the user to change input parameters and test different policies as a great strength as well. Finally the sliders for the sensitivity analysis are also considered as a pro.

### 3.1.2 Cons of the Lumpfish Model in Stella

To the untrained eye, it is not completely clear how the units in Stella function. Re-using a model may prove difficult if many of the variables change, and the user would have to go through significant structure changes. To our best knowledge the edges in Stella have no power, but they can be utilized with if-sentences. Moving a model from Stella requires



exporting the model to another host, as it cannot be hosted on a web server. Using a simulation tool that runs on a Lisp-like programming language is not a benefit, but would rather be better suited with a C++ or Java for example. Lisp [10] is not a commonly used programming language (compared to Java and C++) and therefore offers less support when the user is faced with problems. Using Java or C++ makes it easier to export the code and reuse under different circumstances. Finally the tool used for the model does not have a free software license and does not provide full support unless you have the proper license.

## 3.2 The Lumpfish Model in Orange

In Orange, the user begins with creating the functions as Java Services s/he intends to use for the model and selecting those which are available from the tool. These Java Services are used for the application integration as well. Then the user starts the process development by selecting how many activities s/he wants to use and name them. Next s/he selects what type of activity s/he wants to use, if there is a step that needs human interaction s/he selects a manual activity and so on. After defining which type each activity is the user defines input and outputs for each activity and connects them with edges. If there is a need to have a gateway to wait for a value or decide at runtime where to navigate, the user can select them from the palette in the designer environment. This also applies to the edges between activities, the user can fit them with conditions to make sure they are not taken unless that condition applies. When this is done the user selects an end event or an end activity to finish the model and presses "Generate". The model is generated and the user can fill the activities with the components s/he wishes to use in the development process view. After adding the Java Services to the activities s/he maps the input and output of each function and adds all additional components s/he wishes to use. With this last step the model is ready to run. If the user wishes to change or modify the model s/he has to press generate again after changing it, however if s/he wishes to add functions or analysis it can be done without re-generating the model. Finally if the user wishes to manage hers/his solution, the config for the solution is now available in the web interface, same applies to every model run s/he makes. In figure 3.2 and 3.3 show the lumpfish in Orange and explain its components. In figure 3.4 the process of creating a model in Orange is shown.

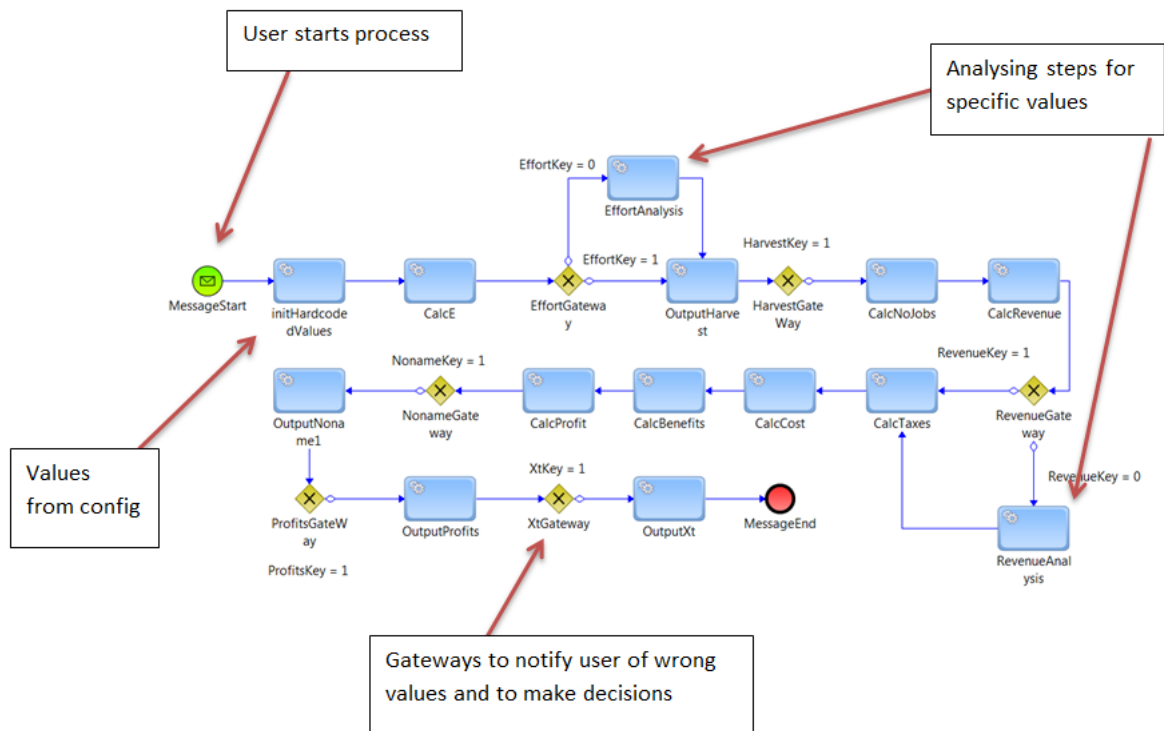


Figure 3.2: The lumpfish model in the Orange

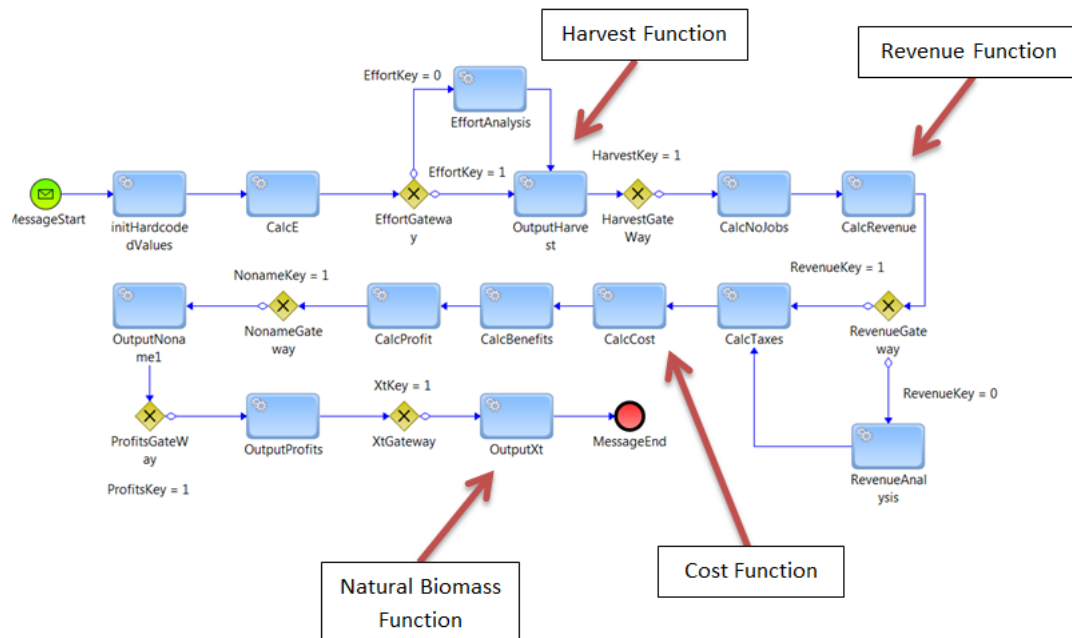


Figure 3.3: Functions in the lumpfish model

### 3.2.1 Pros of the Lumpfish Model in Orange

The model is easy to look at and understand due to a simple design. Reusing the model for similar problems is easy and can be done by either adding activities and new edges or copying the model and replacing its functions. After a function is created it can be re-used as many times as the user sees fit and is accessible from a library in the framework. The application integration is a very powerful tool to use analysis at runtime and the possibility to output any type of file makes it very easy to use the results to do further analysis on the data with data mining. Data mining can also be integrated into the framework with the application integration. Everything is programmed in Java, a common programming language which is available for all platforms and offers a good support. Exporting the framework code is possible, so that makes it easy to move the framework between tools.

### 3.2.2 Cons of the Lumpfish Model in Orange

Although creating a Java Service function does not take long, creating your own library of functions that can be used for all models can take time and there is always a need for someone with a programming knowledge to step in and create the functions. Finally the tool used for Orange does not have a free software license.

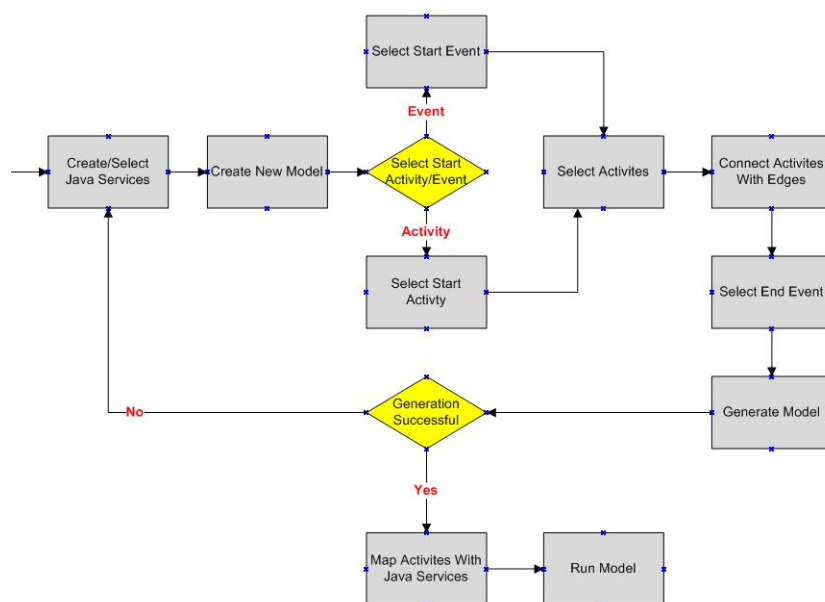


Figure 3.4: Creating a model in Orange

### 3.3 Input Suggestion Model (Pesto)

The Input Suggestion Model (Pesto) was created after we integrated the lumpfish model into Orange, it became evident to us that something could be added to the lumpfish model that had not already been done. Figure 3.5 shows the Pesto model. The lumpfish model works on inputs, which are either estimated or hardcoded by the user and used to calculate functions and return their values. However, there is no way of doing this backwards, that is, inputting the results of the functions and trying to predict the input variables. We decided to create a new model based on the lumpfish model that only accepts the results of the lumpfish model and tries to provide the inputs.

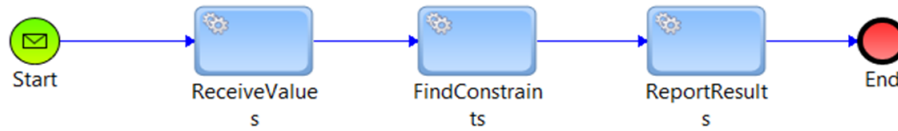


Figure 3.5: The Pesto model

Pesto is based on the same principles as the lumpfish model; it has a start and end event, activities and edges. However it is smaller than the lumpfish model due to the fact that we do not do any additional analysis and all the functions are in the same activity (Find-Constraints). We were able to reuse the events from the lumpfish model, its config and three steps (activities). The functions of the lumpfish model had to be inversed before they were reused. In the first activity "ReceiveValues" Pesto receives the inputs from the document and the config and estimates the same values as the lumpfish models does, then it maps them to the framework solution variables. In "FindConstraints" we use branches, sequences and Java Services (functions) to predict the inputs. In figure 3.6 the functionality of branch is demonstrated to find the value of the variable called *eta*, if the value is already know the step is skipped, otherwise a Java Service is called to find it. Figure 3.7 shows the Java Service for finding *eta*. It has three inputs and if they are all provided, *eta* is returned within a confidence interval and mapped to the output document (*allValues*)

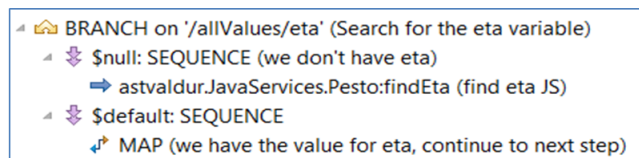


Figure 3.6: Branching for eta and finding its value

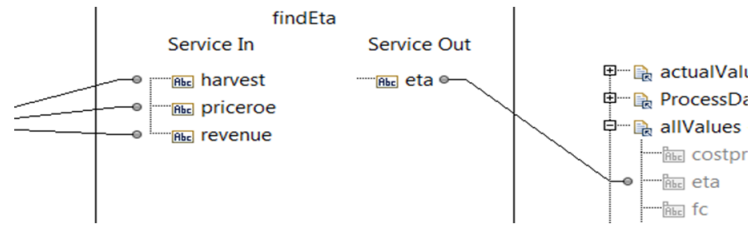


Figure 3.7: Mapping eta to document

The final step of the model is "ReportResults" where the user can use our existing Java Services or create new ones to output the data of the model (allValues) in the desired format. When this activity has finished the model goes to the end event and stops processing.



## **Chapter 4**

# **Analysis of Lumpfish Model using Orange**

In this chapter we explain our analysis of the data from the lumpfish model in Orange and the questions we formed from them. We split the chapter up into six different sections: in the first section we show that the model in our framework is able to get the same output as the model does in the old framework, based on same inputs. The second section explains the questions we formed from the log of data and why they can benefit the research. The third section is for statistical analysis, then in the fourth section we apply our findings to machine learning algorithms and use them to predict the future with time series analysis and forecasting. In the fifth section we talk about the tools we used for our analysis and finally in the sixth section we do a summary and discuss the data used and what we found.

The data we have consists of all fishing trips from March 2008 to August 2012, it is catch specific, holds the name and number of the boat and harbor and has a timestamp for the fishing trip. The fishing season usually starts at the 1st of March, but it is not mandatory that the fishermen start at the beginning of the season. The fishermen have 50 days to fish from the day they start and they do not have any quota.

## 4.1 Consistency of Outputs in Stella and Orange

The aim of the model was to assess the impact of a non-discard policy. In order to do so the lumpfish model was constructed and simulated for 20 years and the two policies were compared with these questions as indicators:

1. What is the profitability margin of the fishery?
2. What is the number of man years in the fishery?

To prove the output of both models were the same we did a run with Stella and changed our model to accept the exact same inputs as Stella used, that is it used the same values that were randomly generated as well. We did an iteration for a 5 year period and compared the outputs. The model uses the output of each iteration as an input for the next iteration.

Variable	Value
costproc	0,98572
eta	0,3
fc	19073,770018
jobsPerTon	0,042328
K	76289,43946
pricelump	346,880962
q	0,00101
r	0,316103
rate	0,05
tax <sub>ate</sub>	0,0
vc	0,4766
randX	0,028612
priceroe	10291
TotalProfitsNonDiscard	0
t	2
dt	1
recruit	6014,56807800453
X	40000
time	1

Table 1: Inputs for the Lumpfish Model

The inputs used for each model are shown in Table 1. Table 2 in Appendix A shows the results for the comparison of data between the model in Stella and in Orange, it resulted in the exact same results.

We also did a run without using the hardcoded values, that is we only used the same inputs for the models. This was done to see the difference between the predefined run and a normal run. For example the variables randX and priceroe, have an upper and lower limit. In our hardcoded run, randX is set to 0,028612, with its lower limit being 0,018980474 and upper limit 0,030935025. Priceroe in the hardcoded run was 10291,



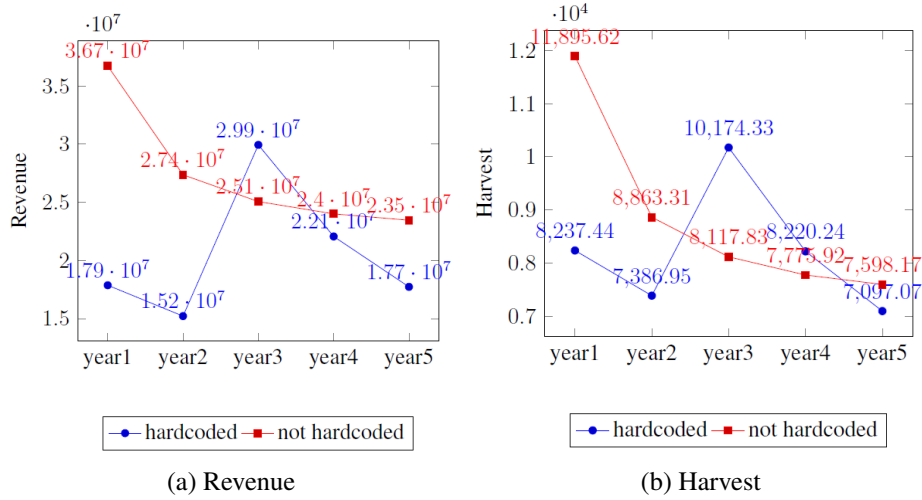


Figure 4.1: Revenue and Harvest with and without hardcoding

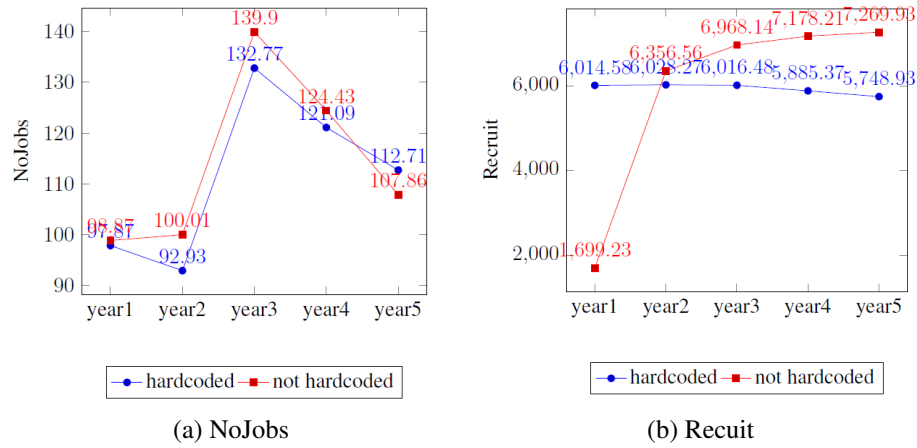


Figure 4.2: NoJobs and Recruit with and without hardcoding

with its lower limit being 6900 and upper limit being 11760. With those two values being found at random, it results in notable change in the overall results from the functions. This is because those two variables are used to calculate the initial variables inserted into the model. For example,  $E$  is calculated with both  $\text{randX}$  and  $\text{priceroe}$ , then  $E$  is used to calculate the harvest, noJobs and cost and they are used to calculate other variables and so forth. The reason we show this run is to demonstrate how different a run can become with stochasticity and without it. We show the differences between selected variables between runs in Figures 4.1 to 4.3, however it should be noted that this is only a single run. We realize that this example does not show any conclusive result, but rather shows how different two runs of the model can be. The results for all the variables in this run can be seen in Table 3 in Appendix A.

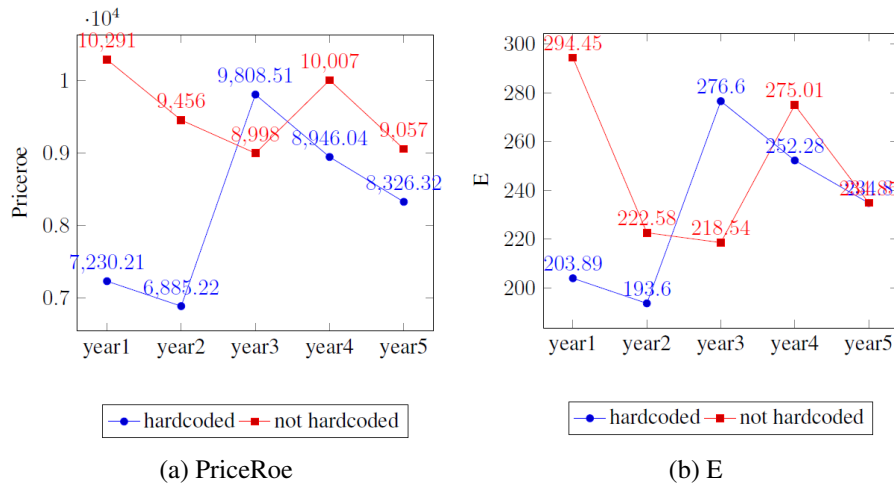


Figure 4.3: PriceRoe and E with and without hardcoding

## 4.2 New Questions

With the application integration in Orange we are able to create a Java Service to read data from a CSV (Comma Separated Values) document. For each row in the document, we map the values to the inputs of our model and run it. At the end of each iteration we add a line to an ARFF (Attribute-Relation File Format) file and a new CSV file. In both cases we have predetermined the structure of the file and tell the pipeline where to map each value. This is done so we will have an easy way of applying data mining. After the files have been created we manually add what type of legislation applies to each year. The years 2008-2010 get the label "Pre", 2011 "Adaption" and 2012 "Active".

From the data we received in the CSV file we did a preliminary comparison of data between years and saw changes between years in the amount of fish caught in the nets, for both lumpfish and other fish. We also saw changes in what month the boats went out to fish, how often they went to fish and the size of some harbors seemed to be growing.

Based on our findings we formed 4 Questions to do statistical analysis on:

- 1-1. How has fishing changed regarding quantities per trip for lumpfish and other fish?
- 1-2. How has the behavior of fisherman changed regarding when they go out to fish?
- 1-3. How has fishing trip frequency changed between years?
- 1-4. How has harbor size changed between years?

### 4.2.1 New Questions Justified

With Question 1-1 we inquire to know more about what is caught in the nets of each boats, and whether it is increasing or decreasing per trip. This is also interesting since there is very limited biological data that can be found of the lumpfish stock and it could give some indications about it. Also keeping track of how much fish is caught gives an indicator about the length of the fishing season.

Question 1-2 is interesting because if all the boats fish in the same month it could result in surplus supply on the markets, causing boats to fish even more than usual to get the wanted revenue.

The reason we are interested in the outcome of Question 1-3 is because by following how many trips each boat makes on average, we can use the findings to predict, based on previous years, how many trips will be made in the following years. Combining that with how much is caught on average we can estimate how much is caught in one fishing season. This can be used as a quota-indicator or to find out how long a fishing season should be.

Question 1-4 looks at the size of harbors between years. We find this interesting since Iceland is split up into six different fishing zones. A boat is never allowed to change its fishing zone, so by tracking boats, we can see where they choose to land their catch. If we see that boats are moving away from some harbor to another that means that the number of job hours are being moved between harbors and can have an effect on the community where the harbor is.

For all these questions we have to assume that the following years will follow a similar trend.

## 4.3 Statistical Analysis

For Question 1-1 we summarized up the total catch for each fishing season and divided them by the number of trips. We see that from the year 2010-2011 the amount of other fish than lumpfish doubles and from 2011-2012 the catch per trip for lumpfish triples. We see that the rise in numbers is around the adaption period for legislation and after it takes effect. The results are shown in figure 4.13.

In Question 1-2 we observe when the boats go out to fish. We are interested in seeing if the adaption period for the legislation and the active legislation has changed the behaviour of

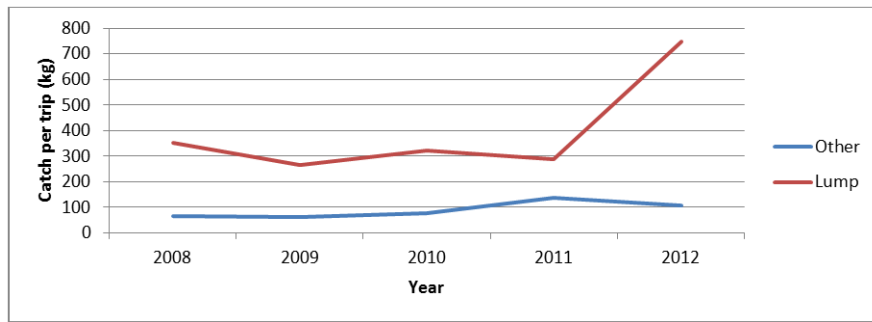


Figure 4.4: Catch per trip on average

fishing trips, that is when the boats go out to fish. What we see clearly is that percentage of all trips made each year is moving from May, June and July to March and April. The results are shown in figure 4.14.

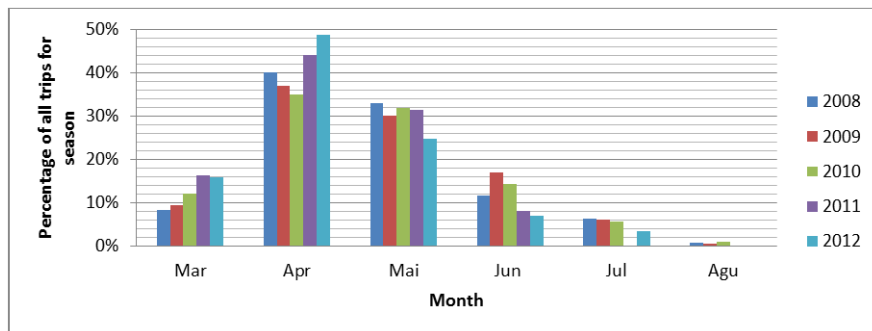


Figure 4.5: Percentage of trips between months

For Question 1-3, fishing frequency between years, we begin with finding out how many boats there are fishing lumpfish around Iceland and the numbers show that since 2010 there have been around 330-340 boats. The results are shown in figure 4.15.

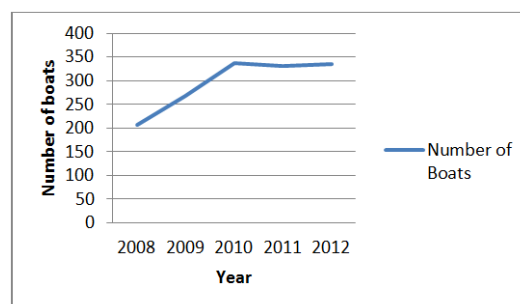


Figure 4.6: Number of boats in Iceland

Then we observe the number of trips made each year from 2008 to 2012. We see a spike in the year 2010 and again in 2012, which is shown in figure 4.16. Since the adaptive period the number of trips has gone down, however there is a 15% increase in trips between 2011 and 2012.

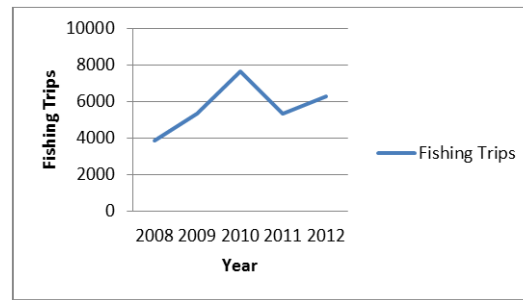


Figure 4.7: Number of fishing trips each year

Finally we divide the number of trips with the number of boats, this is shown in figure 4.17. This gives us a better perspective of how many trips are made. We see that since the legislation boats go out to fish less often, however with what we previously acquired from Question 1 we know that they are also fishing more each trip.

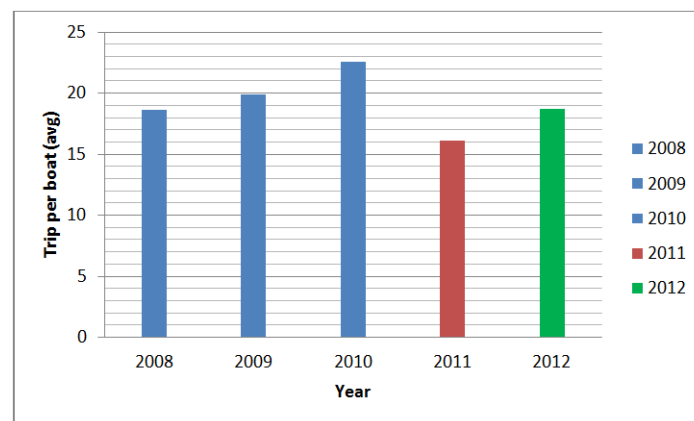


Figure 4.8: Trips per boat on average

Question 1-4 answers how the size of harbors has changed between years and legislation. If we focus on 2012 and compare it to the previous years we see that out of the 60 harbors, 5% of them have doubled or tripled in size and 13% have grown between 20-50%. We also found out that 83% of those harbors belong to 2 out of the 6 fishing zones in Iceland and we do not notice the same decrease of fishing trips in other harbors within these 2 zones. The results are shown in figure 4.18.

## 4.4 Prediction

For prediction we change the questions we had in the statistical analysis. We do this because we are interested in studying Questions 1-1, 1-2 and 1-3 further with time series analysis. The reason question 1-4 was left out was because we felt that we did not have

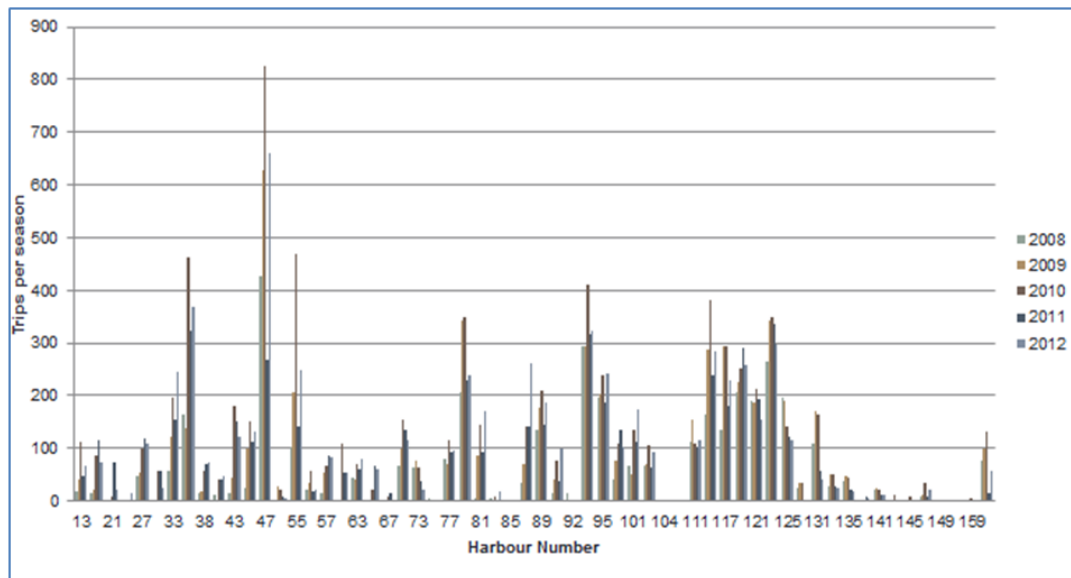


Figure 4.9: Size of harbours between years in terms of number of trips

enough data to predict with. We want to try to predict the outcome for the next years. We look at each Question from different perspectives, where we use parts of the data and all the data. We decided that predicting one fishing season ahead would be sufficient since we have limited amount of data.

The perspectives are:

- #1. The years 2008-2010, to see what we could have expected in 2011. This way we can compare to the actual data
- #2. The years 2011-2012, to see what we can expect of 2013. Here we are only looking at the data since the training period for the legislation and the legislation took effect.
- #3. The years 2008-2012, to see what we can expect of 2013. Here we are using all the data we have.

The reason we are using different perspectives is to try to understand what might have happened without the legislation and the training period for it in 2011-2012. By comparing the data to actual results, 2008-2010 to the year 2011, we hope it will give us a good indicator of what has changed in the behavior of the fishermen. However when we predict for the year 2013 we have no data to compare to, however it will be interesting to compare our results to next year's data to see if our predictions are correct.

The following questions we formed are based on Questions 1-1, 1-2 and 1-3 from 4.2:

- 2-1. What can we expect regarding the amount of fishing caught in the nets for lumpfish and other fish and what can we expect regarding the average catch per trip?
- 2-2. What can we expect regarding the number of fishing trips?

2-3. What can we expect regarding the number of trips per year?

We decided not to research Question 1-4 any further since we felt that we did not have enough data to predict with.

Since the fishing season starts in March and usually has ends in August, we had to fit our data with empty lines so the machine learning tool would know that as well. An example of this is shown in figure 4.19.

Timestamp	TripsPerMonth	SumOtherFish	SumOfLump	SumLumpDivTrips	SumOtherDivTrips
01-2008	0,0,0,0,0				
02-2008	0,0,0,0,0				
03-2008	316,63196,107309,199.9873418,339.585443				
04-2008	1544,145583,631291,94.28950777,408.867228				
05-2008	1267,28544,405073,22.52880821,319.7103394				
06-2008	444,10024,134139,22.57857658,302.1148649				
07-2008	244,7469,70847,30.61065574,290.3565574				
08-2008	29,0,3946,0,136.0689655				
09-2008	0,0,0,0,0				
10-2008	0,0,0,0,0				
11-2008	0,0,0,0,0				
12-2008	0,0,0,0,0				

Figure 4.10: Machine learning file for the year 2008

We split the chapter into different sections for each Question, where we talk about the algorithm used for each questions and the results of it.

#### 4.4.1 Question 2-1: Fishing Amount Per Trip for Lumpfish and Other Fish

For Question 2-1 we used Gaussian Process Regression [11] algorithm (GPR). The reason we decided to use GPR was because we were familiar with it having used it in a previous project. It is a stochastic process whose realizations consist of random values associated with every point in a range of time such the each random variable has a normal distribution. Every finite collection of those random variables has a multivariate normal distribution.

When we predict the year 2011 from the 2008-2010 data, we see that the predictions is that both the lumpfish fishing and other fishing will have a continuous growth. This is shown in figure 4.20.

Based on 2008-2010, 2011 was predicted to have an increase in fishing per trip, for both lumpfish and other fish, as shown in figure 4.21.

When we predict how the year 2013 will look based on the 2011-2012 data, we see a gigantic spike in the sum of lumpfish caught per season, as shown in figure 4.22. There is also a notable increase for other fish.

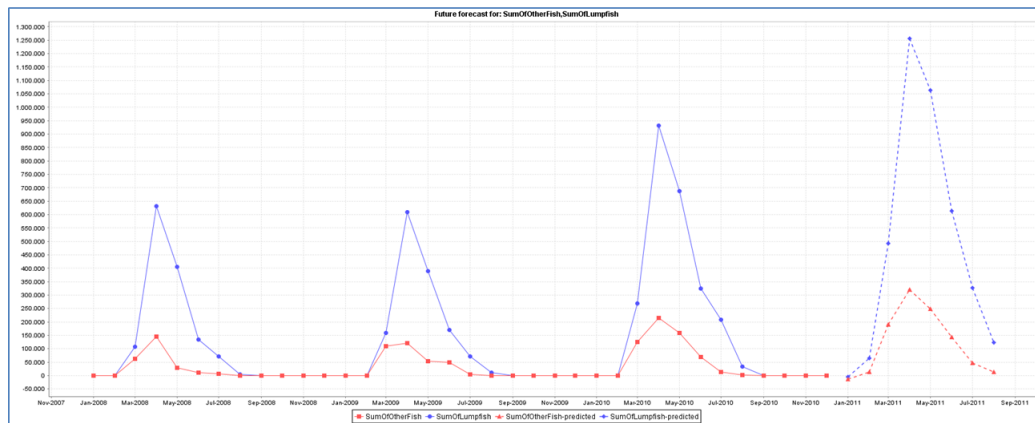


Figure 4.11: Sum of other fish and sum of lumpfish, prediction for 2011 based on 2008-2010

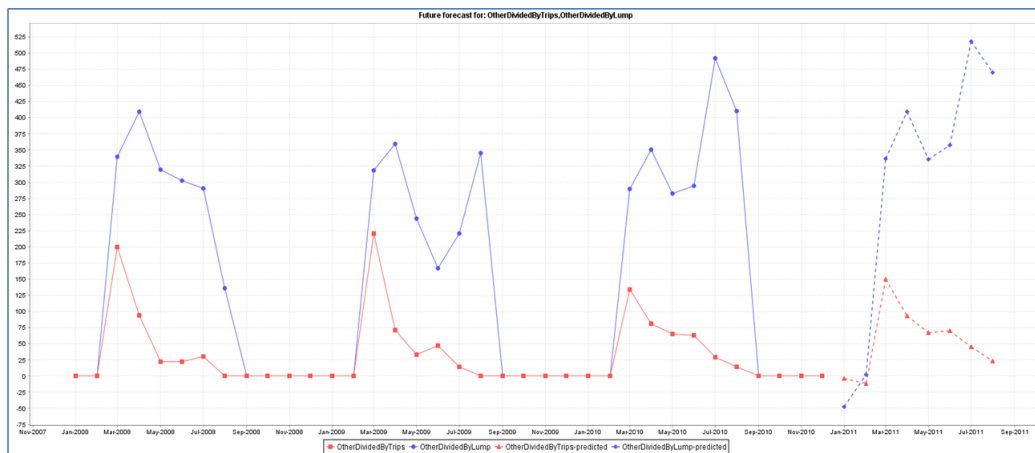


Figure 4.12: Sum of other fish and lumpfish divided by trips, prediction for 2011 based on 2008-2010

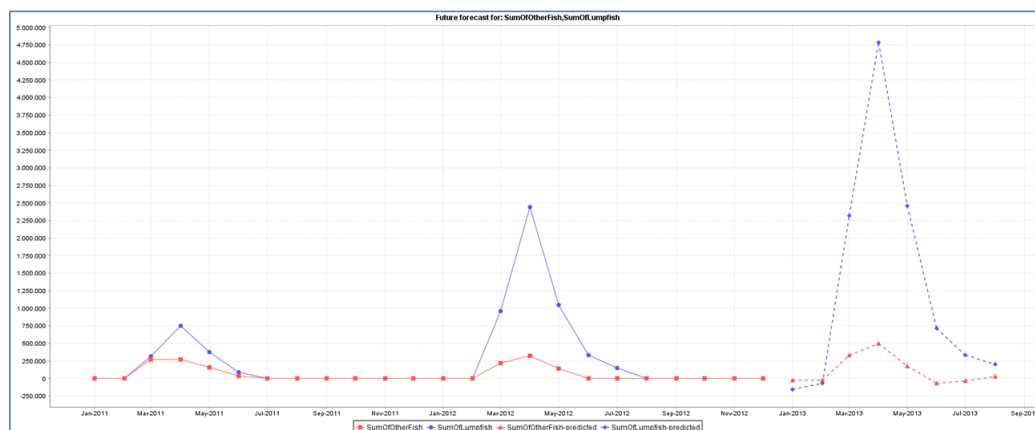


Figure 4.13: Sum of other fish and lumpfish, prediction for 2013 based on 2011-2012

Based on 2011-2012 data, 2013 is predicted to be even bigger per lumpfish trip; however this is not the case for the other fish, as shown in figure 4.23.



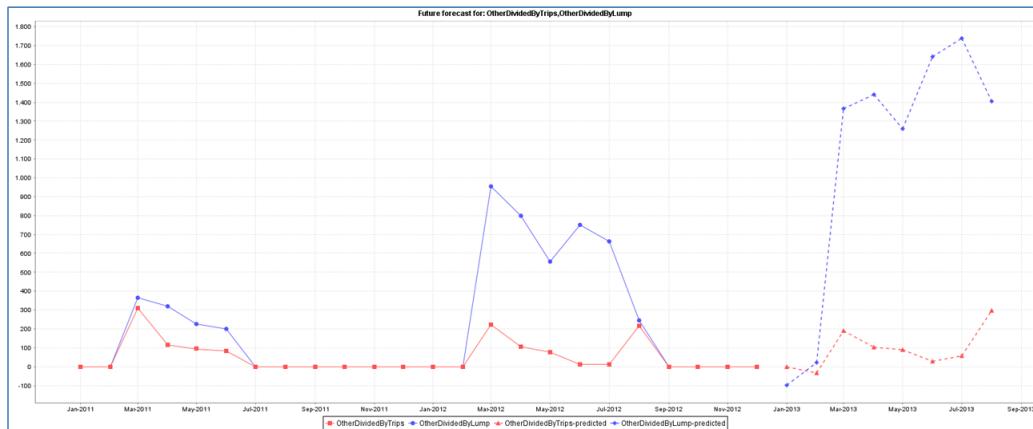


Figure 4.14: Sum of other fish and lumpfish divided by trips, prediction for 2013 based on 2011-2012

When we use all the data to predict for the sum of all trips, we see that there is a possibility that both types become bigger, however it is not expected. We use the GPR with a 95% confidence interval. This is shown in figure 4.24.

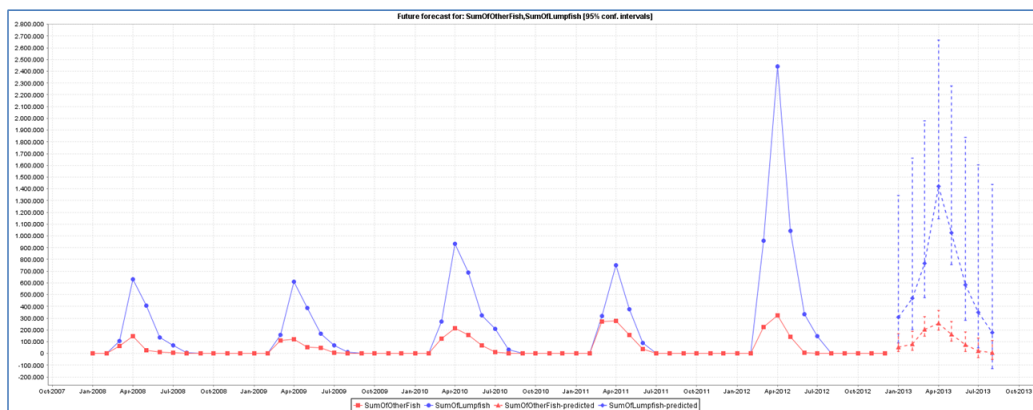


Figure 4.15: Sum of other fish and lumpfish with 95% confidence interval, prediction for 2013 based on 2008-2012

When we use all the data for catch per trip, it is not expected that we will see as big of a spike for lumpfish as the previous year, however the prediction with the 95% confidence interval cannot rule it out, as shown in figure 4.25.

#### Question 2-1 summarized:

**2008-2010:** When we look at the first perspective, we see that 2011 was predicted to be bigger than previous year. However that year we had the legislation training period and in fact the data shows us that in the year 2011 there was a decrease in fishing trips and average catch compared to 2010.

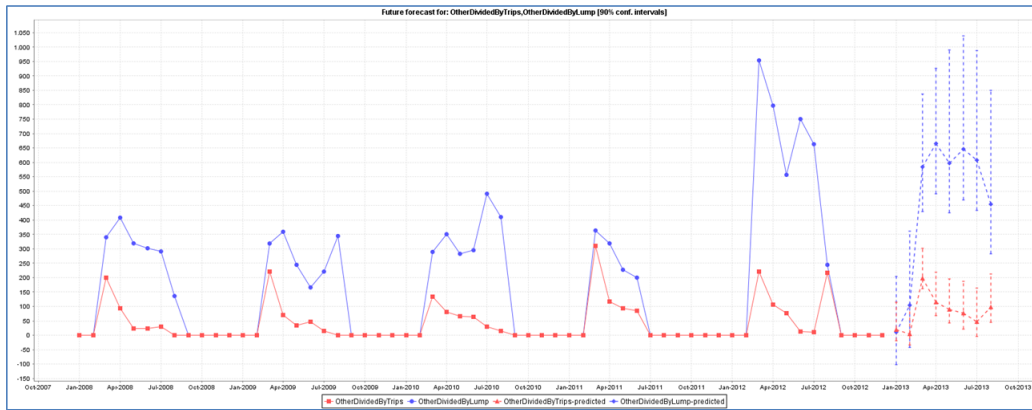


Figure 4.16: Sum of other fish and lumpfish divided by trips with 95% confidence interval, prediction for 2013 based on 2008-2012

*2011-2012:* When we look at the data and try to predict what 2013 would look like, we see that prediction is an even greater increase. However we cannot input into the time series analysis that there is a limit of how much one boat can carry and how many days each boat can fish. Even though these predictions are very interesting and serve a purpose in creating test cases for our model, we feel that they might be unrealistic.

*2008-2012:* When comparing all the data and predicting for 2013 with a 95% confidence interval, we see that the sum of lumpfish 2013 is not expected to be as big as 2012, however it cannot be ruled out. Based on the data for the previous year, if there will be a growth between years it will only be a minor growth. When we look at the catch per trip average we see the same results, predicted that it will go down, but again there is the possibility of a slight rise for both lumpfish and other fish.

#### 4.4.2 Question 2-2: Amount of Fishing Trips

For Question 2-2 we used the Improved smoREG algorithm [12] (IRS). We were familiar with the smoREG algorithm [13] before deciding to use IRS. A college of ours recommended that we should try it out for this specific question and we decided to try it. The changes between the versions are that in IRS [14] locally weighted learning is used to improve the smoREG for regression. Regression being one of the most significant jobs in machine learning and data mining and it is used to forecast numeric target values. It is used to create a model built from a set of training instances with numeric target values and is typically measured by its relative mean absolute error on the test instances. Other algorithms we could have used for this question were naive Bayes [15], k-nearest neighbors [16], linear regression [17] or support vector machine [18]. However the results from

our runs with IRS provided us with the answers we wanted. When we tried using naive Bayes we did not get as accurate results, however they were very similar.

When using 2008-2010 to predict for 2011, we see that 2011 is predicted to be smaller than 2010. However it does not eliminate the possibility that 2011 will be bigger, but it is not expected. This is shown in figure 4.26. The actual values for 2011 are very similar to the prediction made, however March was closer to May and the other way around.

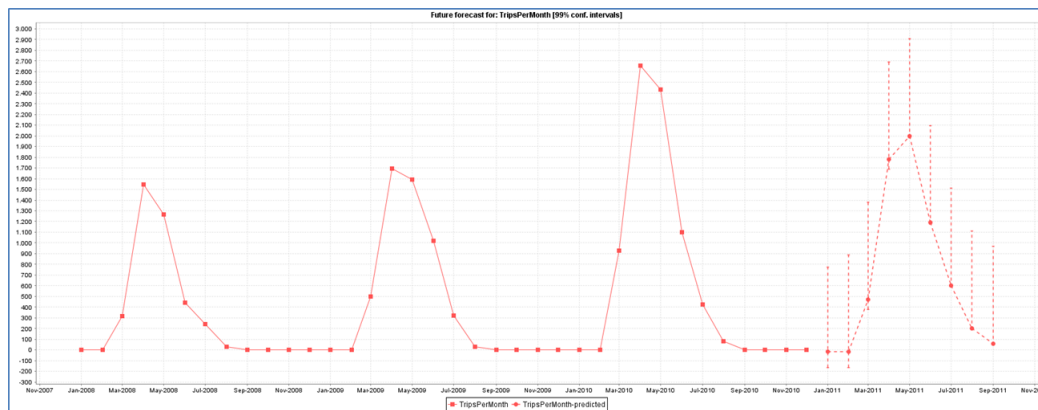


Figure 4.17: Trips per month, prediction for 2011 based on 2008-2010

When predicting the year 2013 based on the data for 2011-2012 we see that the prediction is that 2013 will be somewhere between the amount in 2011 and 2012, as shown in figure 4.27.

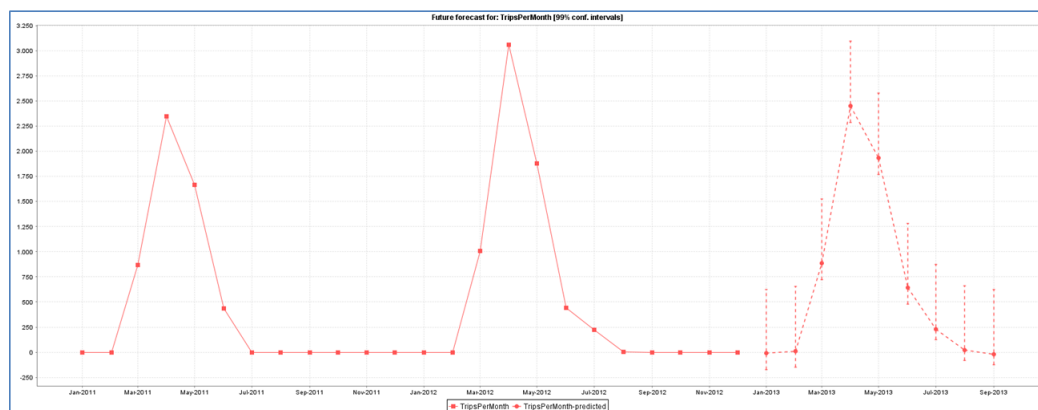


Figure 4.18: Trips per month, prediction for 2013 based on 2011-2012

When predicting based on all the data we have, we see that 2013 is predicted to be smaller for all months compared to 2012. However within the intervals is the possibility that May could become slightly bigger than ever recorded before. This is shown in figure 4.28.

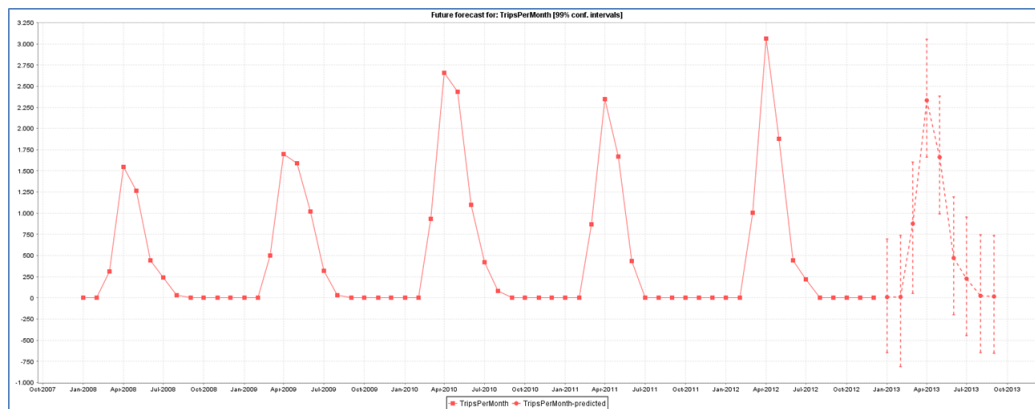


Figure 4.19: Trips per month, prediction for 2013 based on 2008-2012

#### Question 2-2 summarized:

*2008-2010:* We see that the expected trips were predicted to be lower, and that is confirmed by what actually happened.

*2011-2012:* When looking at this perspective we see that it expects 2013 to have fewer trips than year before, but with the possibility of becoming as big.

*2008-2012:* When predicting for the year 2013 based on all the data, we see that 2013 is predicted to be smaller than 2012, however with the exception of May being slightly bigger in 2013.

### 4.4.3 Question 2-3: Amount of fishing trips per year

For Question 2-3 we decided to use GPR again, also since we are working with limited number of data we only predicted this Question based on all the data.

We see that it is expected that there will be a rise in fishing trips for the year 2013, with possibility of the same amount of trips as the year before or a similar spike as in the year 2010. This is shown in figure 4.29.

## 4.5 Tools

In this chapter we list the tools we used for both visualization and prediction. We tried various tools and in the end, we found that these tools listed below suited our needs the best.

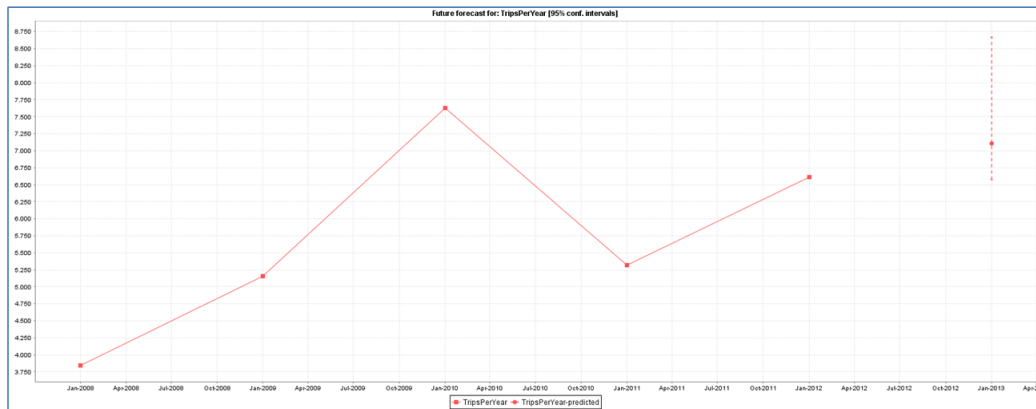


Figure 4.20: Trips per year, prediction for 2013 based on 2008-2012

## 4.5.1 Weka

Weka [7] [19] (Waikato Environment for Knowledge Analysis) is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand and was our choice of tool to use for data mining and machine learning. Weka offers a workbench that contains a collection of visualization tools and algorithms for data analysis and predictive modeling. It also offers graphical user interfaces for easy access. Weka is available free under the GNU General Public License<sup>1</sup>. Since it is implemented with the Java programming language, it is available for almost all modern computing platforms. Weka offers a collection of data preprocessing and modeling techniques and supports several standard data mining tasks, such as clustering, classification, regression, visualization and feature selection. All of Weka's methods are based on the assumption that the data is accessible as a single flat file or relation, where each data point is defined by a fixed number of elements. Weka also offers access to SQL (Structured Query Language) databases using JDBC (Java Database Connectivity) connection and can process the result returned by a database query.

## 4.5.2 timeSeriesForecasting

The Time Series Analysis and Forecasting [8] is an addon offered through the Weka Package Manager, for Weka versions 3.7.3 and higher. It is made by Pentaho, which specializes in Business Intelligence. After installation this environment takes the form of a plugin tab in Weka's graphical explorer user interface. Weka's time series framework takes a machine learning approach to time series by converting the data to a form that standard propositional learning algorithms can process. This is done with removing the historical

<sup>1</sup> <http://www.gnu.org/licenses/gpl.html>

ordering of individual instances by encrypting the time dependency using additional input fields. The additional fields are sometimes referred to as "lagged" variables. After the data has been transformed, regression algorithms can be applied to learn the model. Algorithms offered are multiple linear regression algorithms and non-linear methods such as support vector machines for regression and decision trees with linear regression functions at the leaves.

### **4.5.3 Pentaho Data Integration (Kettle)**

In the Pentaho Data Integration [20] (PDI) also known as Kettle, we seek to use the Spoon [21] perspective. Spoon is a graphical user interface that allows the user to design his own transformations and jobs that can be run with the PDI tools. By adding a plugin step to Spoon it is possible to use the models created with the timeSeriesForecasting addon in Weka and do additional analysis. This also allows the user also to use the same model for different data, instead of having to create a new model for each run in Weka. Additionally PDI provides more detailed data than Weka. PDI offers the user to output the results of each run to different types of files such as CSV, XML, TXT (Text File), ARFF, and accepts and transforms data from them as to use as input for the models, which can be very useful.

### **4.5.4 Excel**

Excel<sup>2</sup> was used to visualize data to answer questions in statistical analysis and create questions for the predictions. We found that visualizing with Excel was better than using Weka's GUI. Also Excel was used to add the legislation values to the machine learning file.

## **4.6 Summary**

First of all we would like to point out that we realize that our questions and findings are not based on much data, however what we are trying to show is that our framework is capable of producing data that can be used for data mining and machine learning, for any type of model that is trying to predict the future. The predictions we made show

<sup>2</sup> <http://office.microsoft.com/en-us/excel/>

either what should have happened in 2011 based on previous years or what we think will happen in 2013. Although some predictions were close to being correct, we take them with prejudice. However it will be interesting to see how close our predictions are when the data for 2013 is ready. In figure 4.30 we show an example of how fishing data for the lumpfish travels through the model in our framework, with the analysis included.

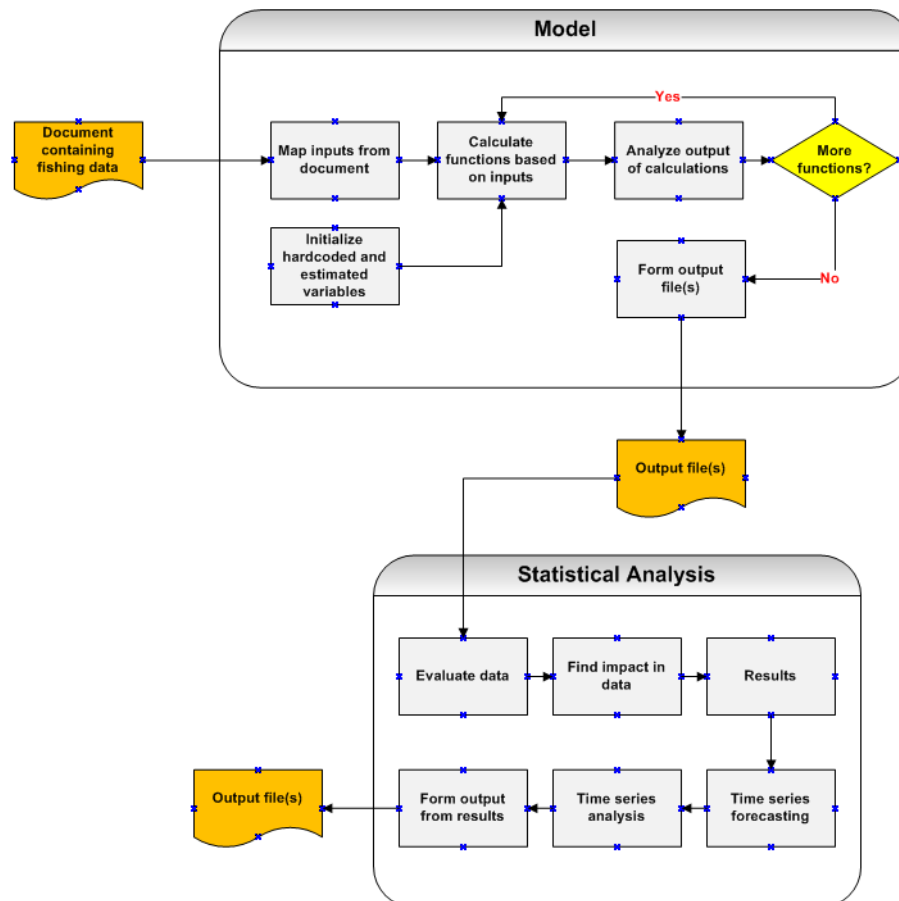


Figure 4.21: Orange: Model and Analysis





## Chapter 5

### Conclusions

We created a powerful framework for building and reforming fisheries system management models. This framework is able to provide the models creator with an environment that can help him in examining and investigating the data on a high level and offers numerous methods to output the data in a structure that is best suited for additional research. The framework offers an easy way to modify and adapt the model to changes. The application integration is an effective instrument in integrating additional tools to the model and to provide it with further analyzes. We are aware that the data we had was very limited and therefore the results from the questions asked might not give concrete information. However we hope to have demonstrated that our framework is capable of reforming and creating any model and can hence benefit anyone in need of reforming her/his models and asking additional questions.



## Chapter 6

### Future Work

In the beginning we set out to create a powerful framework that could create and reform models, analyze them and the data they possess. However this research has spawned some ideas for the continual of the research, both for the model we reconstructed and the framework.

First of all we would like to move our framework to a new tool that has a free software license, but can still provide us with the same capabilities. With this we could make the framework available for everyone to use for modeling and improving.

We are also interesting in using the findings of our models to create an actor based model, a prototype of a "game", similar to the Fish Banks game [22]. The Fish Banks game was developed by Dennis Meadows to allow students to experience the "tragedies of commons" and the way human nature leads to short term decisions and in doing so can destroy resources critical for long term survival. The game offers four different management scenarios and a modeling tool allows the students to determine the optimum management parameters, such as length of fishing season, percentage of area for marine reserves or catch share (depending on the scenario).

We would add to the model a new output file which would create a rule set based on the original output of the model. This rule set would be used to simulate boats in an eco-system. This could be tried with Timed-Rebeca [23] an actor based language with a formal foundation. It is designed in an effort to bridge the gap between formal verification and real applications. Timed-Rebeca can be considered as a reference model for concurrent communication and is based on an operational interpretation of the actor. The user would create the eco-system for Timed-Rebeca, then the rule set from the model would run through it and Timed-Rebeca would output the feasibility of it. With changes in the

system and model we could adapt the model to other type of fish, oil, agriculture or other similar things.

# Bibliography

- [1] Weske, M.: Business process management. Springer (2012)
- [2] Sigurðardóttir, S., Gunnlaugsson, K.: Simulating the impact of policy changes in icelandic lumpsucker fishery. In: Proceedings of the Winter Simulation Conference, Winter Simulation Conference (2012) 377
- [3] Mendes, H., Borges, M.F., Murta, A., Wise, L., Fernandes, P.G., Aschen, M., Balles-teros, M., Brčić, J., Campos, A., Chapela, R., et al.: Ecological, economic, social, and governance indicators for responsive fisheries management systems: an ecofish-man approach. In: 2012 ICES Annual Science Conference. (2012)
- [4] Sterman, J.D.: System dynamics: systems thinking and modeling for a complex world. In: Proceedings of the ESD Internal Symposium, MIT, Cambridge, MA. (2002)
- [5] Costanza, R., Duplisea, D., Kautsky, U.: Ecological modelling on modelling eco-logical and economic systems with stella. *Ecological Modelling* **110**(1) (1998) 1–4
- [6] webMethods, E.: Content type to use. (Document ID: ESTD-EDI-IUG-80SP1-20091204) 142
- [7] Witten, I.H., Frank, E., Trigg, L.E., Hall, M.A., Holmes, G., Cunningham, S.J.: Weka: Practical machine learning tools and techniques with java implementations. (1999)
- [8] Angelis, I.: Data mining pentaho/weka. (2012)
- [9] Parrend, P., Frénot, S.: Classification of component vulnerabilities in java service oriented programming (sop) platforms. In: Component-Based Software Engineer-ing. Springer (2008) 80–96
- [10] Winston, P.H., Horn, B.K.: Lisp. (1986)

- [11] Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research* **6** (2005) 1939–1959
- [12] Shevade, S.K., Keerthi, S.S., Bhattacharyya, C., Murthy, K.R.K.: Improvements to the smo algorithm for svm regression. *Neural Networks, IEEE Transactions on* **11**(5) (2000) 1188–1193
- [13] Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. *Statistics and computing* **14**(3) (2004) 199–222
- [14] Li, C., Jiang, L.: Using locally weighted learning to improve smoreg for regression. In: *PRICAI 2006: Trends in Artificial Intelligence*. Springer (2006) 375–384
- [15] Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine learning* **29**(2-3) (1997) 131–163
- [16] Larose, D.T.: k-nearest neighbor algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining* (2005) 90–106
- [17] Seber, G.A., Lee, A.J.: *Linear regression analysis*. Volume 936. John Wiley & Sons (2012)
- [18] Cortes, C., Vapnik, V.: Support-vector networks. *Machine learning* **20**(3) (1995) 273–297
- [19] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter* **11**(1) (2009) 10–18
- [20] Casters, M., Bouman, R., Van Dongen, J.: *Pentaho Kettle solutions: building open source ETL solutions with Pentaho Data Integration*. Wiley (2010)
- [21] Bouman, R., Van Dongen, J.: *Pentaho solutions: business intelligence and data warehousing with pentaho and mysql*. Wiley Publishing (2009)
- [22] Kozlak, J., Demazeau, Y., Bousquet, F.: Multi-agent system to model the fish banks game process. In: *The 1 st International Workshop of Central and Eastern Europe on Multi-Agent Systems, CEEMAS99, St. Petersburg, DTIC Document* (1999) 154–162
- [23] Aceto, L., Cimini, M., Ingolfssdottir, A., Reynisson, A.H., Sigurdarson, S.H., Sirjani, M.: *Modelling and simulation of asynchronous real-time systems using timed*

rebeca. arXiv preprint arXiv:1108.0228 (2011)





# Appendix A

Table 2:

Variable	Year 1	Year 2	Year 3	Year 4	Year 5
harvest	8237,44	7386,95	10174,33	8220,24	7097,07
New Profit Margin	0,25	0,22	0,3	0,26	0,22
Profit Margin	0,3	0,28	0,35	0,3	0,27
revenue	17867533,44	15213914,14	29938507,21	22061560,37	17727754,63
NewRev	19879001,75	17017703,45	32422938,5	24068827,43	19460760,59
benefits	5440946,51	4249468,64	10364256,53	6707986,99	4774884,43
NewBenefits	4898185,31	3762746,16	9693874,69	6166359,43	4307261,65
cost	12428588,93	10964445,49	19574258,68	15353573,39	12952870,2
newCost	14980816,44	13254957,29	22729061,81	17902468	15153498,94
E	203,89	193,6	276,6	252,28	234,8
priceroe	7230,21	6885,22	9808,51	8946,04	8326,32
total profits non discard	0	4898185	8481753	17274383	22601116
NewResource	0	0	0	0	0
NewBenefitsPV	4898185,31	3583567,77	8792,63	5326733,12	3543594,82
NoJobs	97,87	92,93	132,77	121,09	112,71
NoJobsNewPolicy	448,54	405,6	563,43	469,04	413,11
benefitsPV	5440948,51	4047112,99	9400686,19	5794611,37	3928309,23
Profits each year	5440948,51	4047112,99	9400686,19	5794611,37	3928309,23
recruit	6014,58	6028,27	6016,48	5885,37	5748,93

Table 3:

Variable	Year 1	Year 2	Year 3	Year 4	Year 5
harvest	11895,62	8863,31	8117,83	7775,92	7598,17
New Profit Margin	0,38	0,33	0,31	0,30	0,30
Profit Margin	0,36	0,31	0,29	0,28	0,27
revenue	36725354,16	27363703,94	25062173,64	24006603,64	23457825,73
NewRev	37963263,61	28286058,23	25906949,74	24815799,42	24248523,74
benefits	13334895,2	8435007,48	7230386,52	6677901,18	6390670,82
NewBenefits	14561078,9	9348625,02	8067160,72	7479432,08	7173879,16
cost	23390458,96	18928696,46	17831787,12	17328702,46	17067154,91
newCost	23402184,71	18937433,21	17839789,03	17336367,34	17074644,57
E	294,45	222,58	218,54	275,01	234,85
priceroe	10291	9456	8998	10007	9057
total profits non discard	0	14561079	23464531	30781684	37242699
NewResource	0	0	0	0	0
NewBenefitsPV	14561078,9	8903452,4	7317152,58	6461014,64	5901968,14
NoJobs	98,87	100,01	139,90	124,43	107,86
NoJobsNewPolicy	462,29	419,50	408,98	404,16	401,65
benefitsPV	13334895,2	8033340,46	6558173,71	5768622,12	5257620,7
Profits each year	1334895,2	8033340,46	6558173,71	5768622,12	5257620,7
recruit	1699,23	6356,56	6968,14	7178,21	7269,93





School of Computer Science  
Reykjavík University  
Menntavegi 1  
101 Reykjavík, Iceland  
Tel. +354 599 6200  
Fax +354 599 6201  
[www.reykjavikuniversity.is](http://www.reykjavikuniversity.is)  
ISSN 1670-8539