School of Engineering and Energy

ENG450 – Final Year Engineering Internship

"A report submitted to the School of Electrical, Energy and Process Engineering, Murdoch University in partial fulfilment of the requirements for the degree of Bachelor of Engineering"

Prepared by

Oliver Yeudall

Student No. 30386559

Unit Coordinator

Professor Parisa Bahri

Academic Supervisor

Dr. Gareth Lee

Industry Supervisor

Clifford Lang

Submitted

14-11-2008

ABSTRACT

This is the final report for ENG450 – Final Year Engineering Internship as undertaken by Oliver Yeudall. The specific internship was performed at ISS Group Australia Pty Ltd from the 4th of August until the 21st of November, 2008 where a variety of projects were worked on, providing experience for future careers in industry. Projects included work internally, with Fortescue Metals Group and BHP Billiton. The role of Industry Supervisor was filled by Clifford Lang (previously Warren Jones) and the position of Academic Supervisor was filled by Dr. Gareth Lee. Unit Coordinator was Professor Parisa Bahri.

ACKNOWLEDGEMENTS

The author would like to thank Murdoch University for organising the internship program, especially Professor Parisa Bahri for running the program, and Gareth Lee for acting as academic supervisor on the project. From ISS Group the author would like to thank Frank Zenke for permitting the final year internship to occur at the company, Clifford Lang for acting as industry supervisor, John Knight for providing training services, and Warren Jones for previously acting as industry supervisor. Finally the author would like to thank the staff at ISS Group for assistance performing the internship.

TABLE OF CONTENTS

INTRODUCTION

This is the final report for ENG450 - Final Year Internship. The internship was undertaken to finalise studies at Murdoch University for a degree in Industrial Computer Systems Engineering and Instrumentation and Control Systems Engineering. This specific internship was undertaken at ISS Group Pty. Ltd., where duties were assigned providing experience as a Product Specialist. Over the course of the internship, a variety of activities were experienced, both technical and process related. This provided a range of valuable job skills which would be able to applied in future positions, both with ISS Group and industry in general.

Initially a period of training was needed to train in the use of ISS Group's flagship product BabelFish™, including its related components. Due to the size of the system training was ongoing throughout the duration of the internship, much of it through practical exercises and contract work.

Parallel to initial training in BabelFish, a project was organised which required the use of mathematics training obtained through studies at Murdoch University. In order to fully demonstrate the graphing functionality of BabelFish™ Portal, a project was organised with the goal of deriving a series of pump curves. These would be used for various purposes by ISS Group such as training, sales and general documentation.

Once ISS Group was satisfied that a sufficient amount of training had been given, work was permitted on active projects. These projects involved interacting with clients on-site, such as BHP Billiton and Fortescue Metals Group, and working towards solutions using the BabelFish™ product package.

CHAPTER 1 - ISS GROUP PTY. LTD.

ISS Group was formed in 1995 by Shane Attwell and floated on the Australian Securities Exchange in 2004 [1]. They are a software company which develops and supports industrial software solutions with a focus on Mining and the Oil & Gas sectors of industry. Current clients include corporations such as BHP Billiton, Woodside and Shell. ISS's flagship product is BabelFish™, a data integration and visualisation package which enhances operational management capabilities. ISS also supports BabelFish™ after it has been installed in an operation, consistently improving on the program's architecture.

The internship which was taken at ISS Group was based around the position of 'Product Specialist'. This was a position within the Client Services department, the main roles of which were to interact with clients and support the product as installed in the field. This covers both technical and business related skills such as design, troubleshooting, product testing and general documentation.

BabelFish™ is a framework encompassing many sub-programs. It is a data visualization and management tool for industrial businesses who need to manage large numbers of assets in an easy to understand format. Because it runs as a web service, people can view information on remote sites without the need for client-side installations, reducing setup complexity. It can be used to:

- Monitor field and well equipment
- Track, trend and optimise production
- Manage Asset Integrity
- Monitor activities against operational, environmental, safety and corporate guidelines.
- Monitor operational activities against Corporate Key Performance Indicators (KPIs)[2].

It is currently used by many major entities such as BHP Billiton, Santos, Woodside, BP, Rio Tinto and Saudi Aramco to oversee their global operations.

BabelFish™ and its related components are arranged as shown in Appendix A. In order to understand how BabelFish™ interacts with established processes, each component must be discussed in turn.

BabelFish™ Portal

Portal is a web-enabled software application and acts as the integration layer of the BabelFish™ framework. It can be utilised by general users, where unspecialised personnel are able to view online performance data obtained from multiple site historians. No client side installation is required and can be accessed through most web browsers. An example of Portal can be seen in Figure 3.1, where information is displayed related to the initial training procedure.

BabelFish™ Data Management Server (DMS)

The core of the BabelFish™ framework is the Data Management Server. It acts as a single point where all of a company's data sources are collated and organised into useful information to then be displayed in Portal. It has three main components, which are Data Broker, Data Dictionary and Calculation Engine.

All data sources connect to the DMS through Data Broker. Data Broker provides the single connection point where multiple data sources attach via adaptors. It is a simple solution for the problem of Enterprise Data Integration.

Data Source and Tag information are stored in the Data Dictionary, a hierarchical reference structure which categorises operation assets into one location. It is a powerful tool with many different abilities such as:

- Automated fetching of new data sources
- Asset templating
- Procedurally generated pages

and many more which reduce development costs for clients when setting up and updating the framework.

Calculation Engine provides numerical analysis capabilities for BabelFish™. It has a library of various functions which can use both single-point and historical data sources.

BabelFish™ Software Developers Kit (SDK)

A .NET programming library which enables external development of BabelFish™ applications.

Adaptors

A major strength of BabelFish™ is that it is able to communicate with a wide variety of data sources. Like BabelFish's namesake [3], it allows information from multiple types of source to be translated into a single format. To quote Douglas Adams [3], "if you stick a BabelFish in your ear you can instantly understand anything said to you in any form of language." Data sources which BabelFish™ has adaptors for include [4]:

- Honeywell PHD
- PI
- Exaquantum
- IP21
- Relational (e.g. Oracle)
- SAP
- Maximo
- ODBC/OLEDB
- Web Services

Applications

Along with the basic installation, there are several additional applications produced by ISS Group and additional third parties which are used to enhance the overall functionality of BabelFish™.

<u>SOAP</u> – Microsoft product used for various .NET web services. Pre-requisite for the operation of BabelFish™

<u>Chart FX</u> – Produced by Software FX and is used for advanced charting within Portal.

<u>Operations Daily Reporting System (ODR)</u> – Collates production data from available data sources and presents them within Portal for operator approval. Includes abnormal condition checking abilities and is required for some other applications.

<u>Event Monitor</u> – Application used for monitoring of data conditions in BabelFish. When certain events occur, it is able to notify relevant personnel (Email and SMS), raise work orders and launch other applications.

<u>A-Plus</u> – 'Asset and Production Loss Utilisation System'. Records and classifies all asset downtime events automatically in a hierarchical structure for easy viewing by operators.

Utilities

As part of the basic installation, there are several utility programs which have been produced by ISS. They are used for various functions such as configuring BabelFish™ and debugging. These include:

<u>ISS Enterprise Manager</u> – Administration utility used to manage the Data Dictionary, task scheduler and additional applications. It is used for all setups of BabelFish™ to manage the system at multiple levels, including general Portal administration, debugging and plugin configuration.

<u>Mercury</u> – Logger program used for displaying messages from MS Message Queue. Aids in debugging application and communication issues.

Before new employees of ISS Group are allowed to work on active projects, a period of training had to be undertaken. Initial training in the use of the software for the internship took about 4 weeks to complete, which allowed a level of competency to be attained which permitted work on active projects to occur. In addition, at the time of the internship, the training program was completely new and, as such, input was provided regarding the quality of the training. Once the training program had been finalised, it would become an additional source of income for ISS Group. It would be used to both teach new employees and clients in the use of BabelFish™. This section outlines the training undertaken as part of the internship, along with additional related activities.

The first week of training followed the Basic, Intermediate and Advanced training manuals. Basic training involved learning about the introductory uses of BabelFish™ Portal. It covered navigation, pages, page objects, tags, trends and statistics.

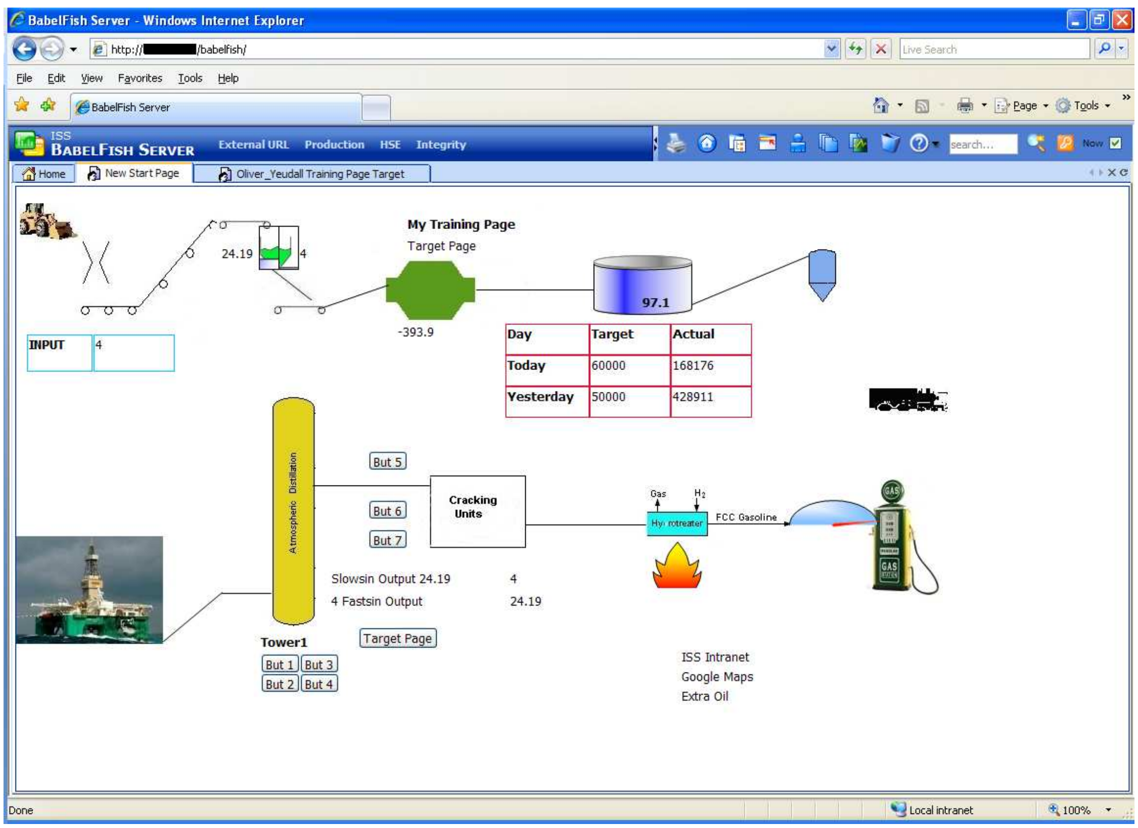


Figure 3.1. *BabelFish™ page built as part of Basic Training*

After the completion of basic training, the knowledge relating to the basics of page building was provided, enabling the construction of Portal pages and their population with configured page objects. Intermediate training expanded on what was learnt in the basic training, such as more detailed information on how to manipulate tags, introducing the concept of iFrames, manipulating page objects through assignable behaviours, constructing performance curve objects and placing them on a page, and how to use chart objects. Advanced training was completed soon after the intermediate. In the advanced training, three major areas were covered; embedding SQL queries on a Portal page, embedding XML on a Portal page, and setting up a combo box to reference both SQL and XML statements to retrieve information from a database.

Once the three initial training programs were completed, a series of exercises related to each level of training were performed. These were assessed to gauge the prowess of those who undertook the training and to finalise the completion of each training level. They involved creating a series of pages related to a well installation which had three oil wells and three gas wells. Pages created and populated for the exercises include a home page showing the field, two pages showing the gas/oil wells and their position in operation, and individual pages for each well.

The next level was the administrator training. Administration in BabelFish™ covered many different areas. These include security, Portal configuration, data source configuration, publishing, page configuration/backup and debugging functions. Unlike the first three levels, initially it was not self-paced. A day's training session was undertaken, followed by a period of experimentation to fully grasp the system after the session. Administrator training involved learning how to configure, manage and troubleshoot the BabelFish™ framework. This not only included Portal, but the Data Dictionary and how it relates to the bigger picture.

Installation of a BabelFish™ server was another important training exercise performed by new employees of ISS Group. Version 2.1.3.6 of BabelFish™ was provided, which was installed on virtual machines as part of the training. 2.1.3.6 is an old version which requires manual configuration of many computer settings in addition to separately installed programs in order to work properly [5]. Much of the installation procedure is handled by the base installer in newer versions of BabelFish™, however performing the process manually provided additional experience which would not have been gained through the automatic installation. Part of the installation required setting up a pair of SQL databases used to store Data Dictionary and Portal reference data. Performing this enabled a greater understanding of how the elements in BabelFish™ interact and how to manually edit them if the need arises.

Like the first three levels, administration training also required the completion of a series of exercises to assess knowledge of the BabelFish™ framework. The exercises expanded on what was achieved in the basic/intermediate exercises by using the Data Dictionary to create templates for oil and gas wells, defining the properties, attributes and relevant renderers. Each well was assigned the relevant template and had their attributes defined via the use of calculated entities to simulate an external data source. In Portal the templates were used to create templated pages, which populated a base page with the relevant properties/attributes for the well being accessed without the need to create individual pages for each well.

CHAPTER 4 - INTERNAL WIKI

To assist the client services and support staff with client assistance, an internal wiki is available which contains common issue resolutions. It is updated on a regular basis to enable efficient error resolution by staff.

Throughout the internship period, several issues were discovered which were resolved. After each had been resolved, both the issues and results were then added to the wiki database for future reference. One of these issues was discovered during the installation training, caused by SOAP services being configured incorrectly. This page was used as a reference when working at Fortescue Metals Group in a later project.

One job related to the wiki was migration of its pages to a new server. A new software package had been implemented at ISS Group which would enable clients to remotely access wiki pages, raise new issues and provide feedback on the process. Because the two wikis were run using different software packages, pages could not be migrated automatically. As a result the migration had to be performed manually by copying and pasting articles from the old wiki software into the new wiki software. Certain articles in the wiki were not appropriate for client access, so they were placed in special sections in the new software.

CHAPTER 5 - SAMPLE PERFORMANCE CURVES

The aim of this project was to assist ISS employees with performance curve documentation/demonstration.

Performance Curves are page objects in BabelFish which allow the display of sampled data against the specified performance of that unit. This unit may be anything with a performance curve such as pumps and separators. In order to set up the object for situations such as training, client demonstrations or job tenders, several sample curves needed to be created. Four sets of sample curves were created, which were each based on real world examples.

1. Set one was a general pump curve (Head vs. Flow) with upper and lower performance limits.
2. The second set was based on the specifications for a single stage pump. This set had three individual curves for Head, Pump Efficiency and Horse Power.
3. Similar to the previous set, the third curve set was with reference to a real pump. It was created to demonstrate the use of the ESP Curve settings within the Performance Curve object.
4. The final set was derived from the performance of a hydrocyclone separator to demonstrate that pumps were not the only process unit which could use the performance curve object.

In order to enter formulas for each curve into BabelFish's performance curve object, six numeric values have to be placed in the format:

$$A, B, C, D, E, F$$

Where the letters are the coefficients of:

$$y(x) = A + Bx + Cx^2 + Dx^3 + Ex^4 + Fx^5$$

Which is the format required by BabelFish™ to plot curves. The general process for deriving each of the curves began by finding a formula $y(x)$ which matched the shape of the original curve. Next the formula was manipulated to fit into the fifth order Taylor series [6]:

$$y(x) \approx y(a) + y^{(1)}(a)(x-a) + \frac{y^{(2)}(a)(x-a)^2}{2!} + \frac{y^{(3)}(a)(x-a)^3}{3!} + \frac{y^{(4)}(a)(x-a)^4}{4!} + \frac{y^{(5)}(a)(x-a)^5}{5!}$$

Where *a* was chosen to provide the best fit with the original formula. A sum of least squares calculation was used to minimize the difference between the two curves.

The first two derived sets of performance curves used a combination of performance limit curves and discrete sampled data to simulate the measured degradation of each pumps' impeller over time. As time progressed, the sampled data would progressively illustrate the pump becoming worse in its performance. The driving formula for the sampled data was arranged so that is would "reset" every January to simulate a scheduled maintenance of the pump, bringing its performance back to the upper limit. This is shown in the following figures, where older data is represented by lighter points and newer data is shown by darker points.

Sample Set 1 – Generic Pump Curve

This example is based off of a general pump curve relationship rather than a specific model of pump. Using the original formulas, the performance curves look like such:

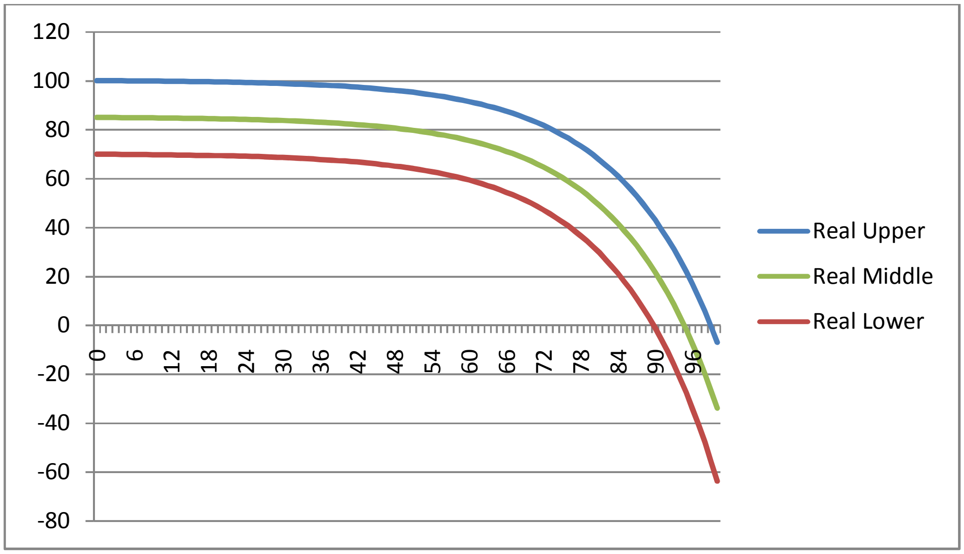


Figure 5.1. *Sample Performance Curve for generic pump curve*

Where the x axis represents Flow Rate and the y axis represents Head Pressure. The upper and lower curves represent the upper and lower allowable limits of the pump's performance, whereas the middle curve represents the sampled data.

Original Formulas:

$$y_{Upper} = \frac{1}{5}\left(1 - e^{2x\pi/100}\right) + 100$$

$$y_{Middle} = \frac{1}{4.5}\left(1 - e^{2x\pi/100}\right) + 85$$

$$y_{Lower} = \frac{1}{4}\left(1 - e^{2x\pi/100}\right) + 70$$

Once these had been evaluated around $a = 52$ and placed in the 5th order Taylor series, the polynomial equivalents were determined to be:

$$YU = (-4.28\text{E}-08)x^5 + (7.73\text{E}-06)x^4 + (-6.66\text{E}-04)x^3 + (2.84\text{E}-02)x^2 - 0.66x + 105.96$$

$$YM = (-4.8\text{E}-08)x^5 + (8.59\text{E}-06)x^4 + (-7.4\text{E}-04)x^3 + (3.16\text{E}-02)x^2 - 0.73x + 91.62$$

$$YL = (-5.35\text{E}-08)x^5 + (9.66\text{E}-06)x^4 + (-8.33\text{E}-04)x^3 + (3.55\text{E}-02)x^2 - 0.83x + 77.45$$

Using these coefficients, the original curves became:

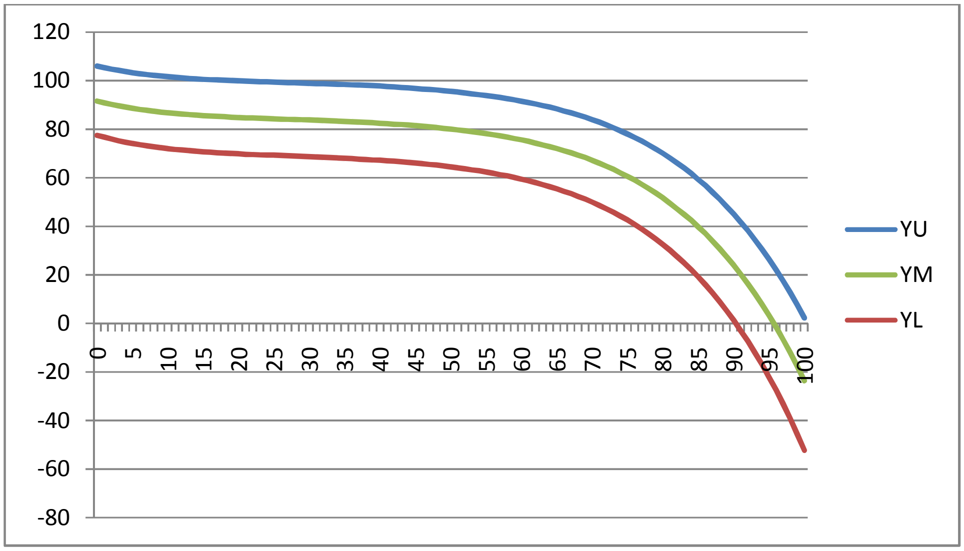


Figure 5.2. *Taylor series transformation of Sample Set 1*

When using this as an example for pump performance curves within BabelFish™, YU and YL were plotted as the upper and lower limit curves. A version of YM was used to simulate sampled data from a historian.

The derived coefficients used for limit curves were:

YU = [105.96,-0.66,2.84E-02,-6.66E-04,7.73E-06,-4.28E-08]

YL = [77.45,-0.83,3.55E-02,-8.33E-04,9.66E-06,-5.35E-08]

To simulate sampled data, the Performance Curve object requires two sequences of data; one for the X value of the sample and one for the Y value of the sample. For this curve the X axis was Flow and the Y axis was Head which references Flow as a dependant variable. The driving formula for the Flow tag used was:

40+TAG(CALC.RANDOM.NORMAL)/5

Where CALC.RANDOM.NORMAL was a random value with a uniform distribution between 0 and 100. The formula for Head referenced this tag, and was:

(1/4.5)*(1-EXP(2*PI*TAG(FLOW)/100)) +100
+((JANUARY(TIME)-TIME)/850000)

Where TAG(FLOW) was replaced with the tag for the flow produced by the pump.

Using these two data tags, several factors are indicated to the viewer:

- The pump's standard operation is a flow rate between 40 and 60.

- As the year progresses the performance of the pump will deteriorate.
- It is shown that the pump undergoes maintenance every January. At that point it will be operating outside its lower operational specification.
- Once it has been repaired, it returns to near optimal operation.

After the curves and data have been set up to be drawn by the Performance Curve object, the graph shown in fig 5.3 is generated:

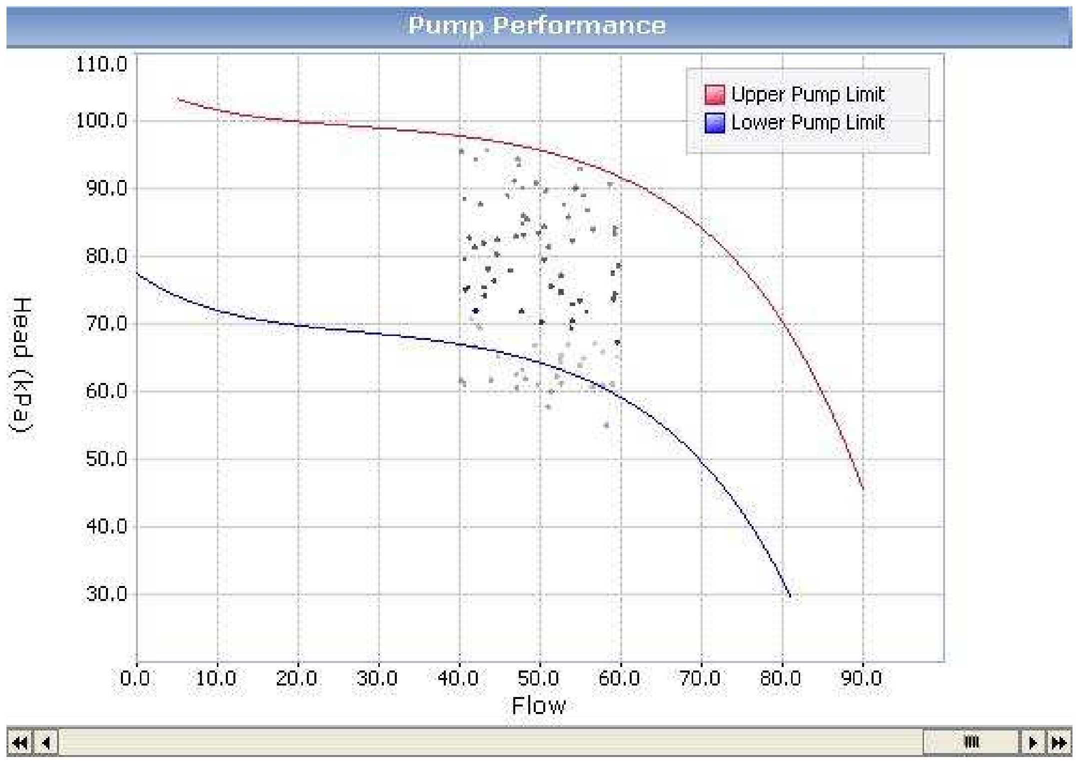


Figure 5.3. *Generic Performance Curve example rendered in BabelFish™*

If there is sampled data plotted on a Performance Curve object, it is organised by age. Darker points indicate newer data, where lighter points indicate older data. For this plot, the data was based around the previous year's samples from October 2007 to October 2008.

Sample Set 2 – Real Submersible Pump Example

This next set is based on the performance curves of an actual pump:

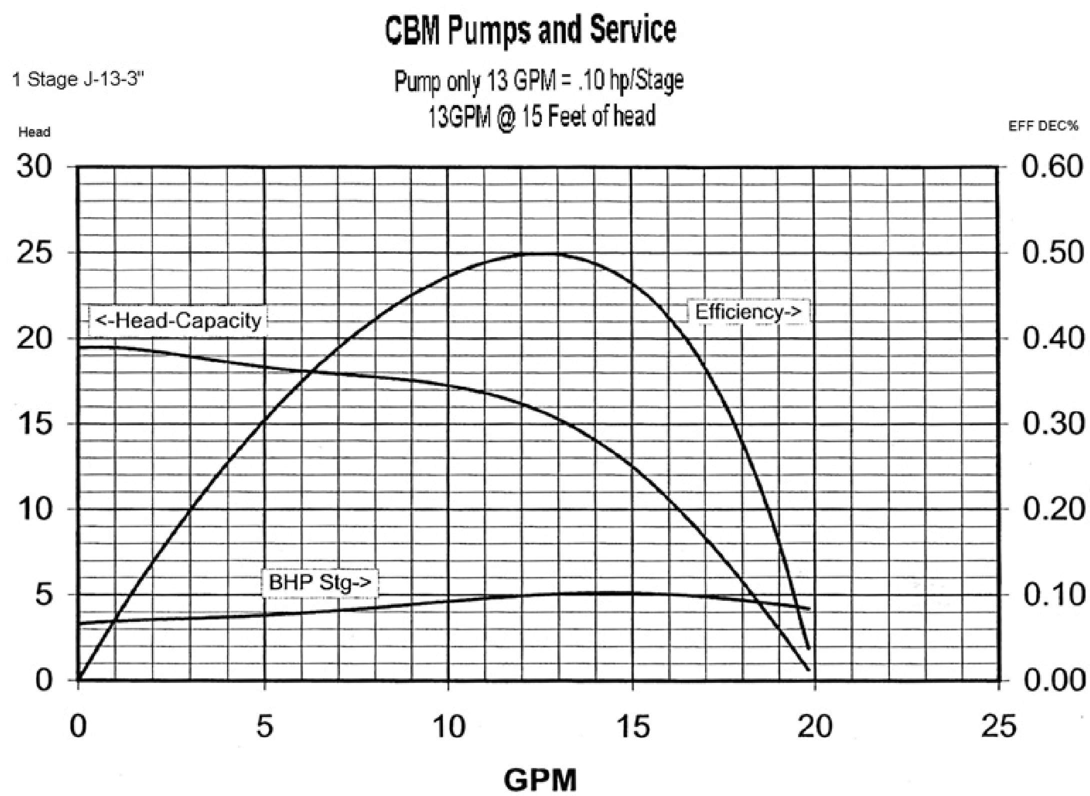


Figure 5.4. *Pump curve specifications of a single stage electric submersible pump* [7]

This graph in fig 5.4 contains three individual curves for a single stage ESP; Head, Efficiency and Horse Power. For the BabelFish™ examples, all three curves were analysed separately. Using the same method outlined for sample set 1, the base equations for the three curves were determined through experimentation to be:

Head Capacity Equation:

$$y(x) = \frac{5}{2\sqrt{2\pi}} e^{\frac{(x-1)^2}{-8}} - \frac{1}{4} e^{\frac{\pi x}{15}} + 18.75$$

Efficiency Equation:

$$y(x) = \frac{1001}{1000} - e^{\frac{-2\pi x}{100}} - \frac{1}{1000} e^{\frac{\pi x}{10}}$$

Horse Power Stg Equation:

$$y(x) \frac{1}{4(4.2)\sqrt{2\pi}} e^{\frac{(x-14.5)^2}{-2(4.2)^2}} + \frac{1}{1000}\left[\left(1 - e^{\frac{\pi(x-14)}{17}}\right) + \left(1 - e^{\frac{-\pi(x-14)}{17}}\right)\right] + 0.08$$

When plotted, the three equations became:

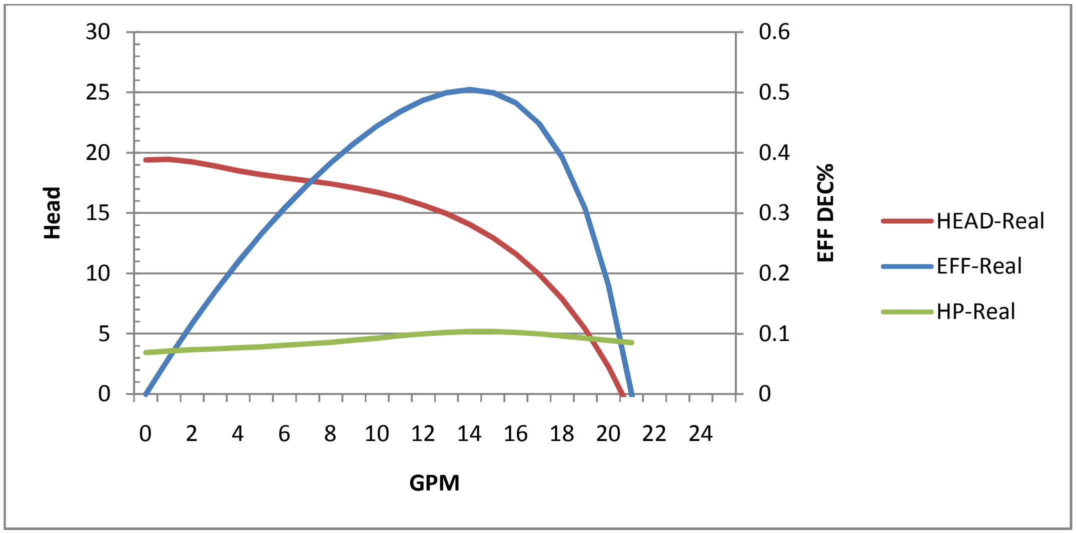


Figure 5.5. *Taylor Series equivalent of sample set 2*

Using these equations, the 5th order Polynomial Equivalents were found:

$$y_{HEAD} = (-1.28\mathrm{E}-05)x^5 + (5.27\mathrm{E}-04)x^4 + (-1.16\mathrm{E}-02)x^3 + 0.116x^2 - 0.7286x + 20.04$$

Evaluated around a = 11 for best fit

$$y_{EFFICIENCY} = (-8.04\mathrm{E}-07)x^5 + (3.10\mathrm{E}-05)x^4 + (-5.36\mathrm{E}-04)x^3 + (3.29\mathrm{E}-03)x^2 + 0.0371x + 0.04948$$

Evaluated around a = 13.27 for best fit

$$y_{HP} = (6.02\mathrm{E}-07)x^5 + (-3.24\mathrm{E}-05)x^4 + (6.18\mathrm{E}-04)x^3 + (-5.03\mathrm{E}-03)x^2 + 0.0185x + 0.0574$$

Evaluated around a = 13.27 for best fit

When it came to plotting the curves in BabelFish™, each curve was plotted on a separate Performance Curve object. To simulate performance limits, the Taylor polynomial was used as the upper limit for each, with the lower limit using the same equation with a y-axis offset. Because of this, the six sets of coefficients entered into BabelFish™ were:

HEAD – Upper Limit = [20.04,-0.7286,0.116,-1.16E-02,5.27E-04,-1.28E-05]

HEAD – Lower Limit = [15.04,-0.7286,0.116,-1.16E-02,5.27E-04,-1.28E-05]

EFFICIENCY – Upper Limit = [0.04948,0.0371,3.29E-03,-5.36E-04,3.10E-05,-8.04E-07]

EFFICIENCY – Lower Limit = [-0.25052,0.0371,3.29E-03,-5.36E-04,3.10E-05,-8.04E-07]

HP – Upper Limit = [5.742,1.849,-5.03E-01,6.18E-02,-3.24E-03,6.02E-05]

HP – Lower Limit = [2.742,1.849,-5.03E-01,6.18E-02,-3.24E-03,6.02E-05]

An issue was discovered while plotting Horse Power. Ranges of plotted performance curves are defined using integer values, so ranges which are less than 1 cannot be plotted. As such plotting Horse Power on a Y-axis range of 0-1 resulted in a graph which was unreadable. In order to compensate, the coefficients for the Horse Power curves were scaled by a factor of 100.

When plotting the simulated sampled data against each curve, the X and Y axis values were required. Because all three Performance Curve objects referenced the same X axis values (i.e. a measured GPM liquid flow), a single tag was referenced in all three equations:

CALC.RANDOM.SMALL+8+((JANUARY(TIME)-TIME)/20000000)

This equation would cause the overall flow rate to reduce over time. Using this X-axis tag, the three Y-axis values for Head, Efficiency and Horse Power were derived using the following calculations:

Head

-1*(1.28*POWER(10,-5))*POWER(TAG(GPM), 5)

+(5.27*POWER(10,-4))*POWER(TAG(GPM), 4)

-(1.16*POWER(10,-2))*POWER(TAG(GPM), 3)

+(0.116)*POWER(TAG(GPM), 2)

-(0.7286)*POWER(TAG(GPM), 1)

+19+((JANUARY(TIME)-TIME)/6000000)

Efficiency

-1*(8.04*POWER(10,-7))*POWER(TAG(GPM), 5)

+(3.10*POWER(10,-5))*POWER(TAG(GPM), 4)

-(5.36*POWER(10,-4))*POWER(TAG(GPM), 3)

+(3.29*POWER(10,-3))*POWER(TAG(GPM), 2)

+(0.0371)*POWER(TAG(GPM), 1)

+0.0174+((JANUARY(TIME)-TIME)/100000000)

Horse Power

100*((6.02*POWER(10,-7))*POWER(TAG(GPM), 5)

-(3.24*POWER(10,-5))*POWER(TAG(GPM), 4)

+(6.18*POWER(10,-4))*POWER(TAG(GPM), 3)

-(5.03*POWER(10,-3))*POWER(TAG(GPM), 2)

+(0.0185)*POWER(TAG(GPM), 1)

+0.0474+((JANUARY(TIME)-TIME)/1000000000))

After setting up three individual Performance Curve objects to plot Head, Efficiency and Horse Power, these three graphs were obtained and are shown in figures 5.6 to 5.8:

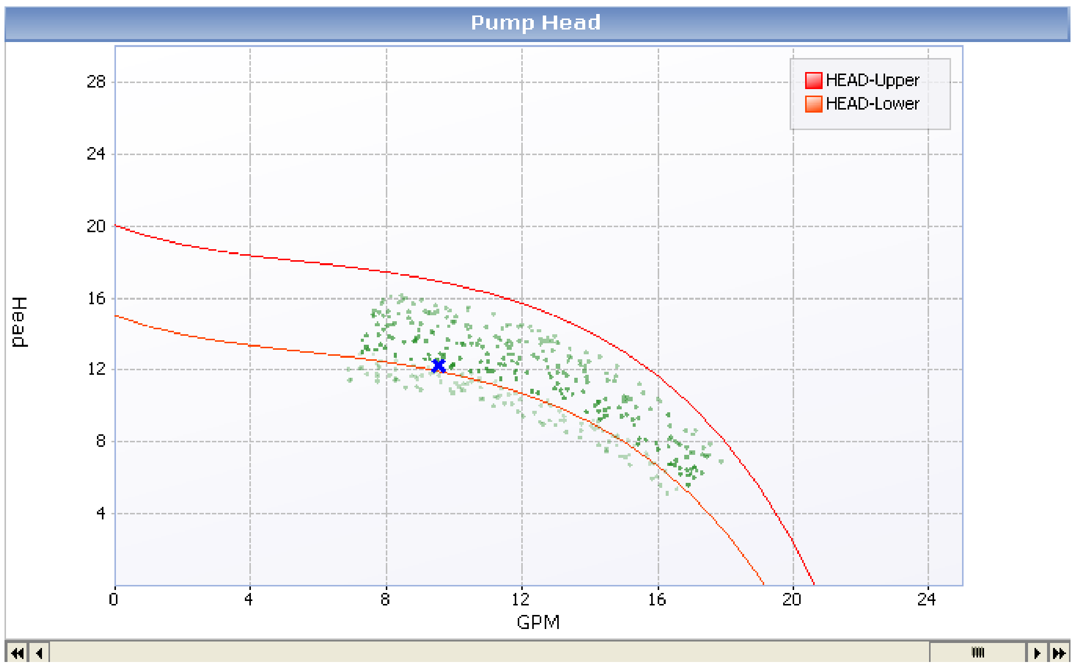


Figure 5.6. *Sample Set 2 Pump Performance – Head vs. Flow rate*

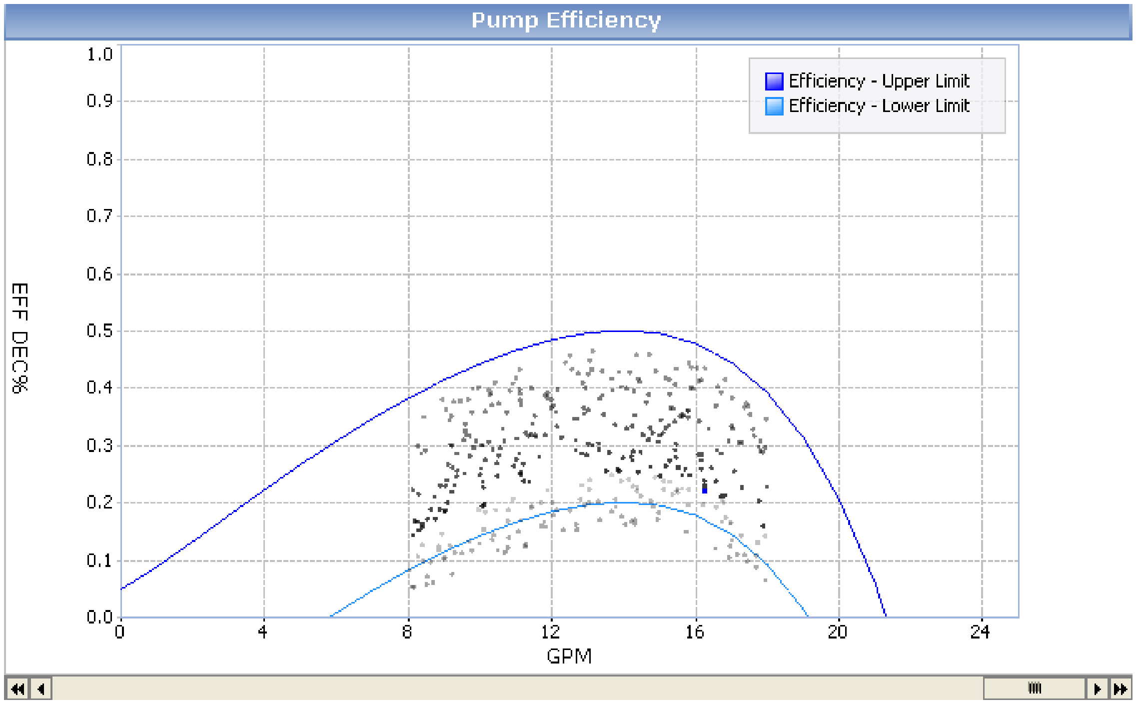


Figure 5.7. *Sample Set 2 Pump Performance – Efficiency vs. Flow rate*

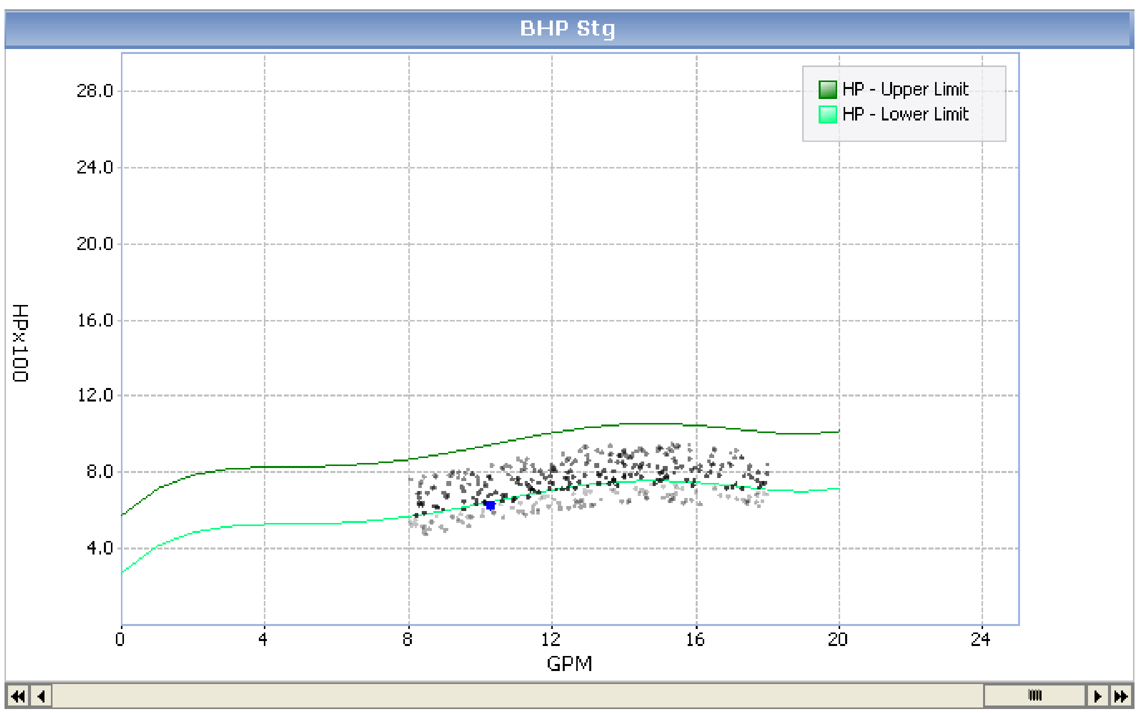


Figure 5.8. *Sample Set 2 Pump Performance – Horse Power vs. Flow rate*

Sample Set 3 – Dedicated ESP Curve

Sample Set 3 was used to illustrate the functionality of the ESP curve object. The ESP curve object behaves in a similar fashion to the General Performance Curve, with a few differences. An ESP curve object is only able to accept one series of polynomial curve coefficients, however it is able to automatically plot that curve with reference to multiple operational frequencies. It does this with reference to the following function:

$$y(x) = \left(A + Bx + Cx^2 + Dx^3 + Ex^4 + Fx^5\right)\left[stages\left(\frac{frequency}{60}\right)^2\right]$$

Where *stages* was equal to the number of impellers the ESP unit has and *frequency* was the rated operating speed. *A* to *F* are the normal coefficients for a performance curve. This example was based on the performance curve of the pump in figure 5.9:

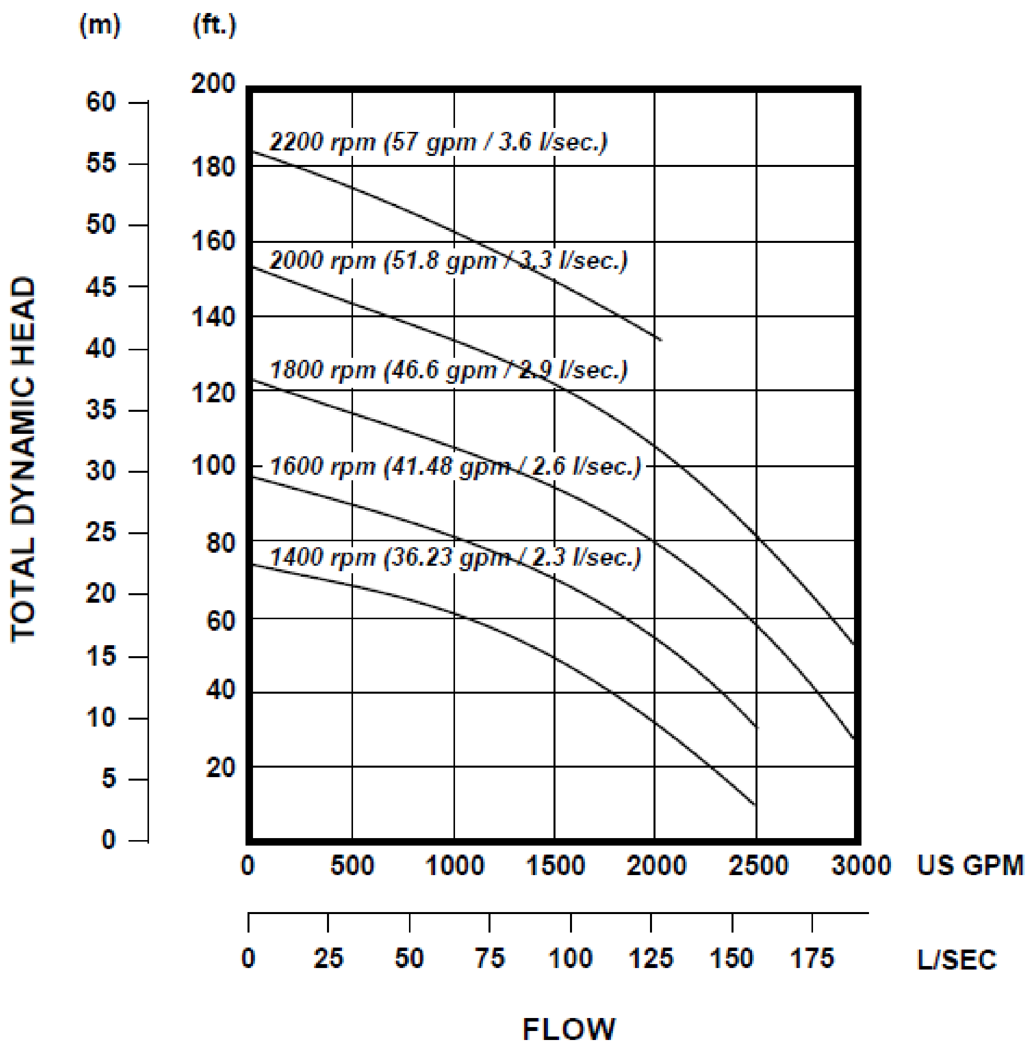


Figure 5.9. *Performance curves of a pump at varying operational frequencies* [8]

The base equation derived from this diagram was at 1800RPM (60Hz).

$$y_{1800RPM}(x) = \frac{-1}{10} e^{\frac{\pi(x+870)}{2000}} - \frac{x}{60} + 123.1$$

With the 5th order Taylor series equivalent being

$$y_{1800RPM} = (-3.298\text{E}-16)x^5 + (1.424\text{E}-12)x^4 + (-3.795\text{E}-09)x^3 + (3.882\text{E}-06)x^2 + (-2.007\text{E}-02)x + 123.437$$

In order to scale this to 30Hz as opposed to 60 for the ESP curve object, the polynomial became:

$$y_{HEAD} = (-4.122\text{E}-17)x^5 + (3.559\text{E}-13)x^4 + (-1.897\text{E}-09)x^3 + (3.882\text{E}-06)x^2 + (-4.014\text{E}-02)x + 493.748$$

Using this equation and the frequencies given in the original graph, the following ESP curve object was created:

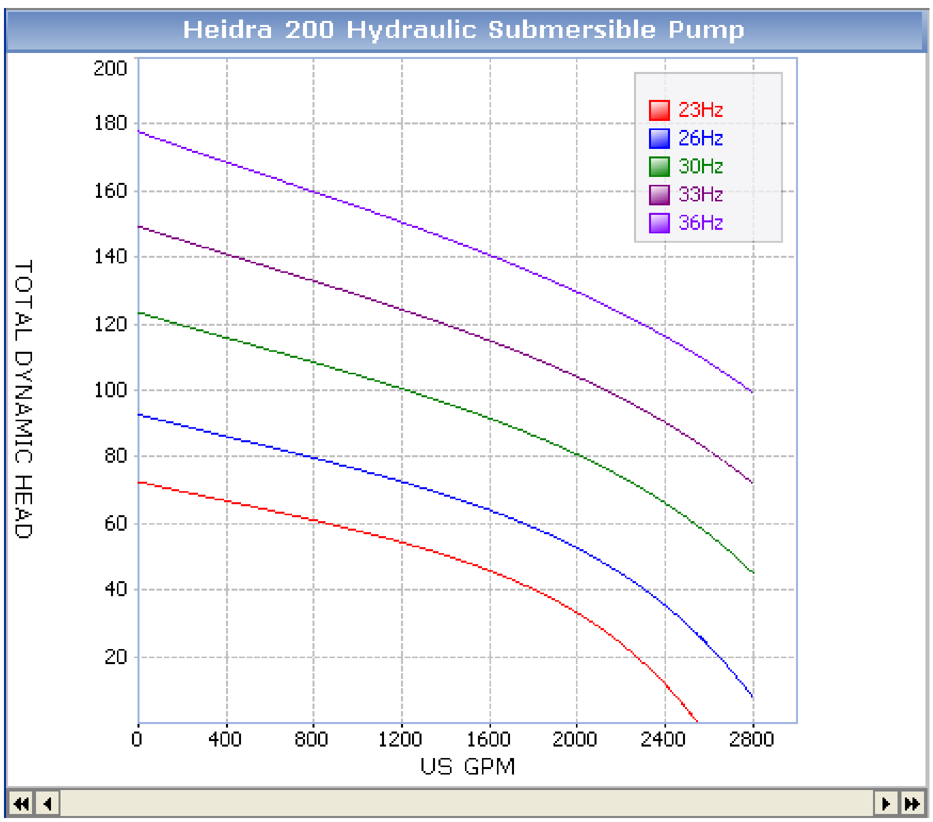


Figure 5.10. *Sample Set 3 Pump Performance at different operation frequencies*

Sample Set 4 – Non Pump Example

Sample Set 4 was different in concept to the previously constructed sets. It was derived to illustrate that the Performance Curve object was not limited to pump curves, but a wide variety of different curves. This set of curves was based on the performance of a hydrocyclone separator manufactured by SPINIFEX, used to process industrial runoff.

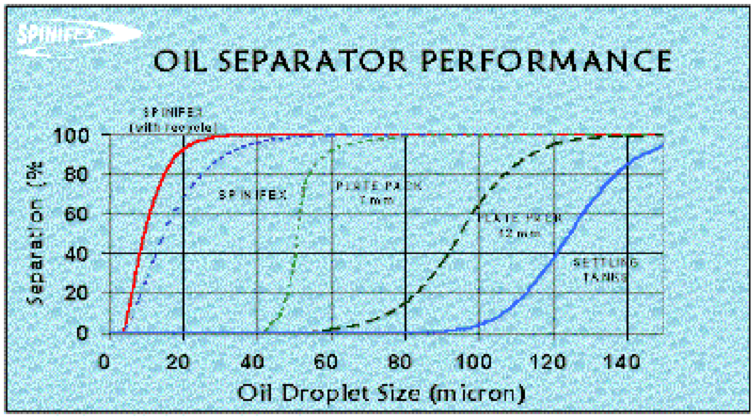


Figure 5.11. *SPINIFEX hydrocyclone separator performance compared to generic separators* [9]

From left to right, the curves are:

- SPINIFEX SP2000 (With Recycle)
- SPINIFEX SP2000 (Single Pass)
- High Quality Plate Pack (7mm)
- Budget Plate Pack (12mm)
- Simple Gravity Settling Tank

Developed equations for the first two curves matched the actual curves, however the others were not as accurate. Because the curves were being produced for example purposes, it was deemed that an equation that would match the general shape would be acceptable. The derived equations were:

SPINIFEX SP2000 (With Recycle)

$$f(x) = 100\left(1 - \left(1 + \frac{(x-2)}{4}\right)e^{\frac{-(x-2)}{4}}\right) - 17.56394\left(1 + \frac{x}{0.3}\right)e^{\frac{-x}{0.3}}$$

SPINIFEX SP2000 (Single Pass)

$$f(x) = 100\left(1 - \left(1 + \frac{(x-2)}{7}\right)e^{\frac{-(x-2)}{7}}\right) - 4.949129\left(1 + \frac{x}{0.35}\right)e^{\frac{-x}{0.35}}$$

High Quality Plate Pack (7mm)

$$f(x) = 100\left(1 - 0.5e^{-(x-45)} + 4e^{\frac{-(x-45)}{2}} - 4.5e^{\frac{-(x-45)}{3}}\right) - (2.7649E+192)(1 + 10(x+10))e^{-10(x+10)}$$

Budget Plate Pack (12mm)

$$f(x) = 100\left(1 - \left(1 + \frac{(x-70)}{11}\right)e^{\frac{-(x-70)}{11}}\right) - 478251.242e^{\frac{-(x-10)}{8.4}}$$

Simple Gravity Settling Tank

$$f(x) = 100\left(1 - \left(1 + \frac{(x-96)}{12.9}\right)e^{\frac{-(x-96)}{12.9}}\right) - 8934480.17e^{\frac{-(x+8)}{10}}$$

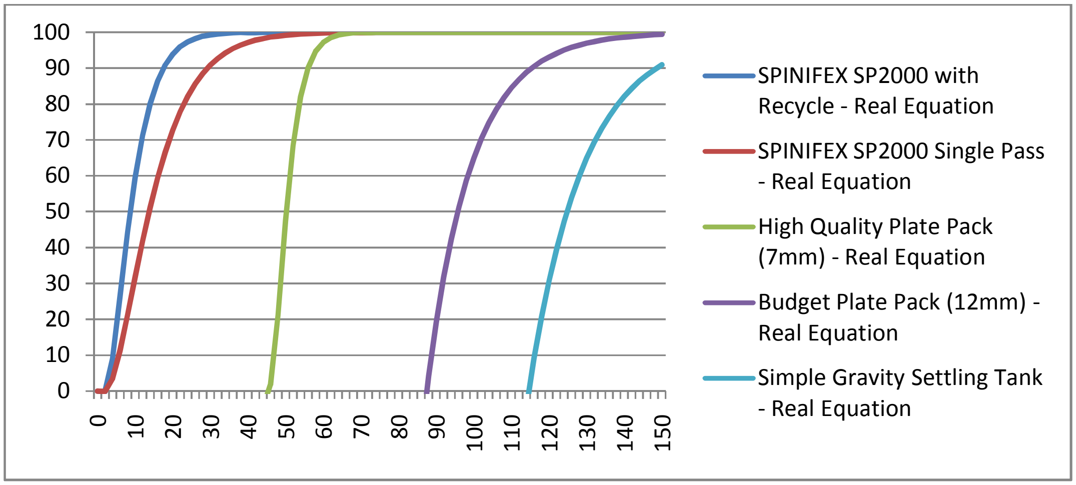


Figure 5.12. *Taylor Series approximations of sample set 4*

5th order Polynomial Equivalents:

SPINIFEX SP2000 (With Recycle) – Evaluated around $a = 32$

$$y = (1.575\text{E}-06)x^5 + (-2.926\text{E}-04)x^4 + (2.211\text{E}-02)x^3 - 0.852x^2 + 16.825x - 36.588$$

SPINIFEX SP2000 (Single Pass) – Evaluated around $a = 52$

$$y = (1.232\text{E}-07)x^5 + (-3.771\text{E}-05)x^4 + (4.711\text{E}-03)x^3 - 0.301x^2 + 9.898x - 34.400$$

High Quality Plate Pack (7mm) – Evaluated around $a = 65$

$$y = (1.491\text{E}-05)x^5 + (-5.093\text{E}-03)x^4 + 0.697x^3 - 47.864x^2 + 1646.561x - 22619.884$$

Budget Plate Pack (12mm) – Evaluated around $a = 81.62$

$$y = (5.251\text{E}-08)x^5 + (-4.662\text{E}-05)x^4 + (1.527\text{E}-02)x^3 - 2.396x^2 + 184.013x - 5520.071$$

Simple Gravity Settling Tank – Evaluated around $a = 105.8$

$$y = (1.288\text{E}-09)x^5 + (-1.942\text{E}-05)x^4 + (1.099\text{E}-02)x^3 - 2.416x^2 + 242.299x - 9248.376$$

Using the coefficients of these equations, the final Performance Curve object displays as in figure 5.13:

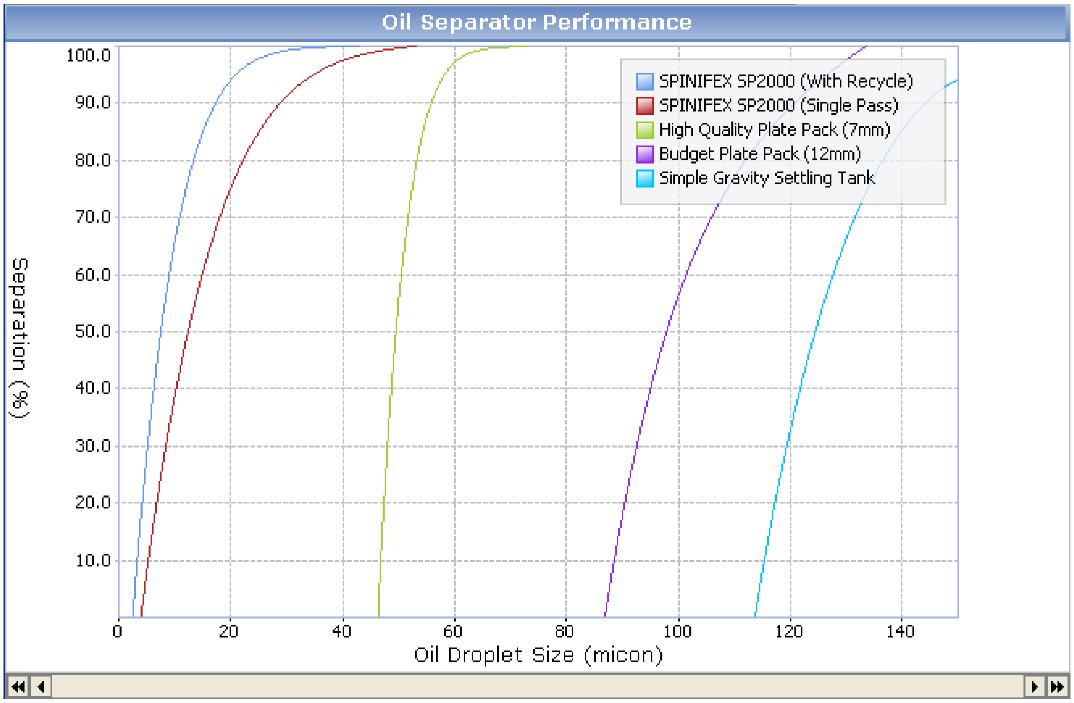


Figure 5.13. *Performance Curves of separators as rendered in BabelFish™*

Nickel West is one of ISS Group's major projects. BabelFish™ is used to collate data from all five of its operation sites situated in various locations in Western Australia, to be used by the corporate headquarters back in Perth. Part of the activities taken as part of the internship involved working on BHP Billiton's Nickel West initiative, initially as a training exercise but later in a support role for the installed systems.

Training

As a training exercise, a reconstruction of the BabelFish™ setup of the Leinster Nickel Operation's (LNO) installation was requested as per the Detailed Design Document for Horizon 1 of the project [10]. The basic page layouts were provided, with the key task being to populate each with calculated tags. In the actual system, the tags represent references to LNO's PHD data historian. Because of this the task was to simulate the process with calculated tags using the same names as the tags in the actual system. If the original tag was a measured process variable, the driving formula would be:

$$Mean + RANDOM(-Deviation, +Deviation)$$

This would mean the outputted value of the tag would deviate from the mean by a random number between $-Deviation$ and $+Deviation$. The mean was selected based on information provided in the Detailed Design Document, where images of the final product were used as a basis for each mean. If the original tag was a Boolean status indicator, then the equation would be:

$$IF(RANDOM(10) > x, 1, 0)$$

This would generate a random number between 0 and 10. If that number was greater than x, the output would be 1, else 0. x could be any number between 0 and 10, chosen to vary the weighting of each outcome.

After all the main training was completed, permission was given to apply that knowledge to projects relating to the actual Nickel West setup.

Tag Transfer

During the fifth week of work at ISS, a project was participated in involving the manual transfer of tags between a testing server and a production server. During this procedure, a bug was discovered which prevented all the tags on a given portal page to display. It was discovered that the error was caused in two situations; when there were too many tags of a certain type (limits on values) or faulty data source references. This error was passed on to the development team for inclusion in a future patch.

Horizon 1 - BabelFish™ 2.1.5 Site Acceptance Testing (SAT)

There are several versions of BabelFish™ in use with the setup at Nickel West using 2.1.3, an older version of the software. Version 2.1.5 is the latest release and as a result Nickel West requested an upgrade. Because Nickel West is the first client to have the latest version, test procedures needed to be defined. Version 2.1.5 was installed on a development server and configured to retrieve data from site historians in the same manner as the production server. All the Portal pages were also exported to the development server. SAT would be performed on the development server to make sure that 2.1.5 was ready for deployment on the production server. Testing involved several distinct stages which are summerised below.

BabelFish™ Installation Verification

Confirming that the core elements of BabelFish™ and relevant web-services were working. Involved following the general procedure outlined in the BabelFish™ Installation Guide [5].

Applications/Adaptors Installation Verification

Confirming that the applications and adaptors used at Nickel West worked with the new version of BabelFish™.

Global Security Settings

Ensuring that the global security settings were the same on the development server as on the production server.

Performance

Assessing the overall performance of BabelFish™ 2.1.5 as a web-service. This consisted of such tests as access time, load testing and infinite loop handling.

Page Display

Validating that Nickel West pages on the new version of BabelFish™ displayed the same as the old version.

Page Tag Assignments

Checking that the tags used and referenced on each Nickel West page were the same in both versions, including tags assigned to embedded trends, status indicators and level indicators.

Page Conditions

Making sure that all discrete-state indicators changed status when their driving tags were returning specific values from historians.

Site Security Settings

Confirming that the user group permissions worked correctly under version 2.1.5 for each Nickel West site.

Entity Assignment

Validating that all site entities were returning the same data as on the old version of BabelFish™.

Each of these sets required the construction of a variety of test cases. These began with testing the higher level sections, like the BabelFish™ core program, then moving on to more export validation style tests to confirm the proper export of pages from the old server. These tests were created and performed manually by ISS Client Services using Test Director at the request of BHP Billiton.

Horizon 2 – Loss Asset Utilisation

An addition to the base upgrade of BabelFish™, an expansion of the actual system was requested by BHP Billiton. This involved upgrades to existing pages, new pages/templated pages and entity hierarchies. Activities undertaken included preparation for Factory Acceptance Testing (FAT), troubleshooting server issues and assisting in the development of the Detailed Design Document. This project had not reached completion by the submission date, however it is scheduled to continue up to and beyond the end of the internship time period (21st of November).

One of ISS Group's major clients is Fortescue Metals Group (FMG). FMG uses BabelFish™ to oversee its West Australian operations. As an internship project, with the assistance of ISS personnel on-site, a BabelFish™ server migration was performed. This involved setting up BabelFish™ and all the components relevant to this system on a new production server, with all the settings and configuration data stored in a separate SQL database. This was performed over a period beginning on Friday the 26th of September until Friday the 3rd of October with time billable to FMG. Installation of the software was performed under the supervision of FMG's System Administrator, Michael Pryor.

Installation

The SQL database for BabelFish™ is kept on a separate server to BabelFish™ itself to increase overall system performance. Before work on the project had commenced, the SQL database had already been migrated to a new server by Michael Pryor.

Installation and configuration of the relevant software was performed according to the procedure outlined by Product Consultant Andrew Chew. Because the install was based on an existing server, the installation involved more steps than a standard install. These additional steps included:

- Copying/modifying configuration files from the old production server to the new production server.
- Running scripts to update connection parameters for plug-ins and BabelFish™ adaptors.
- Copying custom VML (Vector Markup Language) and XML code which has been developed specifically for FMG.
- Setting up appropriate DSN connections.
- Configuring additional applications used within Portal for use on the new server.

Testing

Once the main installation procedure had been completed, the project moved on to the testing phase before the new server could be commissioned. This involved following the Quality Assurance procedures set down by ISS to properly validate a fresh BabelFish™ installation [5].

After verifying the installation of BabelFish™, all the individual Portal pages had to be tested to make sure that all links worked correctly, all the tags displayed properly, and all the tags showed the same data as on the old production server. This also verified the correct install and configuration of the adaptors and plug-ins, many of which were used to generate page objects.

While performing these tests several issues with the BabelFish™ install were discovered and are listed below.

<u>Issue 1</u>

In order for web users to access BabelFish™ Portal (as hosted on the web server), the reference SQL databases are required to have specific database owner names and passwords. When the SQL database was copied from the old SQL server to the new one, all the owner passwords had reset. This caused a variety of problems when attempting to access Portal with a web browser. Depending on which combination of passwords were incorrect, various errors were returned by Internet Explorer, such as “This page cannot be displayed”, “You do not have permission to view this page” and “Service Unavailable”.

In addition to the SQL database security settings, certain folders on the BabelFish™ server are required to have custom access permissions in order to allow the BabelFish™ software to function. Once these permissions were properly set up access to the BabelFish™ server through a web browser was enabled, solving many other related issues.

Issue 2

In Portal, entity hierarchies from the Data Dictionary can be accessed by a user through the Plant Hierarchy Tree toolbox. Entities cannot be modified in this view, however they can be used for general reference and page building purposes. After the installation the Plant Hierarchy Tree was unable to be displayed, returning the error “Input string was not in a correct format”. It was discovered that this error could be caused by two potential problems:

- The config file for the BabelFish™ Data Dictionary points to the machine name instead of ‘localhost’.
- An improper installation of SOAP services

As SOAP services had been left out of the original installation, that was deemed to be the issue. After installing SOAP the Plant Hierarchy Tree was able to be accessed by Portal users.

Issue 3

One of the verification tests to be performed on a BabelFish™ server is to access the URL http://**[server_name]**/BFDD and follow three hyperlinks. Each hyperlink runs a query which tests the web services connection to the BabelFish™ Database. When these tests were run, an error was returned stating “Expected input string of type xml but returned html”. This issue was solved by the installation of SOAP services, however it has also been known to occur when the improper xml parsers have been installed.

CONCLUSION

This was the Final Report for ENG 450 – Final Year Engineering Internship. Much was learnt throughout the internship period, both technical and business process related. A variety of projects were worked on, ranging from initial training to testing, which provided a wide range of experience for future careers in industry. Projects worked on as part of the internship were outlined and the main details of which were discussed.

BabelFish™ Terminology

BabelFish™	Data management and visualisation framework for industrial processes. Able to organise data from many 3rd-party data sources for operational management and process optimisation.
Chart	Similar to trends but uses the third-party Chat FX plugin to create advanced graphical representations of data.
Data Dictionary	Hierarchical organisation of all the data elements which comprise an industrial operation. The three major components are Entity Hierarchies, Data Sources and Templates
Entity	Asset instance in the Data Dictionary which can represent either a physical item (pump, valve ect) or a logical grouping (e.g. production facility, oil field). Types of entities include data source references and calculation references.
ESP Curve	Subsection of the Performance Curve object which is configurable for the unique performance characteristics of electronic submersible pumps.
iFrame	Embedded object in Portal pages which are able to render external pages within the BabelFish™ framework.
Page	Pages in Portal act as individual pages on a website. XML based, they can be displayed on PCs with Internet Explorer or web-enabled mobile devices such as PDAs.
Page Objects	Item which are able to be placed on a Portal page to be viewed by anyone who accesses that page. Examples include hyperlinks, images, tags and embedded trends.
Performance Curve	Page object which is able to display static performance curves on the same graph and compare them to historic data from a specified period of time.
Plant Model Tree	Gives general users basic access to the entity hierarchy tree of the Data Dictionary from within Portal. Cannot be used to modify the hierarchy, but can be used for referencing data sources.
Renderer	URL reference within Portal which can be configured to display Portal pages in specific ways, such as opening a word document, populating a page with filtered data and opening tag details in a non-standard way. Can be attached to templates, menu items or URL hyperlinks.
Statistics	Time series statistics of a tag in Portal over a given time period, such as standard deviation, mean, maximum and minimum.

Tag	Displayable page object in portal which can either display calculation results or act as a reference to a data source set up in the data management server.
Template	Data Dictionary templates are used to define the general properties and attributes a grouping of similar entities will have, for example a well template to define what the characteristics of all wells in an oil field are such as depth, yield and status.
Templated Page	Page which uses the properties and attributes of a template. When accessed via an entity which uses the base template, that page is automatically populated with the data relevant to that base entity.
Trend	Time series graph in Portal. Can be displayed as a separate page or embedded within another Portal page.

Calculation Engine Terminology

CALC.RANDOM.SMALL	A random value between 0 and 10 using a uniform distribution.
CALC.RANDOM.NORMAL	A random value between 0 and 100 using a uniform distribution.
IF(EXPR, A, B)	Evaluates an IF statement. If the case EXPR is true, then A is returned, otherwise B is returned.
JANUARY(TIME)	Time since 1st of January of the current year.
POWER(EXPR1, EXPR2)	The value of EXPR1, raised to the power of EXPR2.
RANDOM(EXPR)	Random value between 0 and EXPR using a uniform distribution of values.
RANDOM(EXPR1, EXPR2)	Random value between EXPR1 and EXPR2 with a uniform distribution of values.
TAG(EXPR)	Fetches the value of EXPR, where EXPR is replaced with a tag reference.
TIME	Current system time in seconds.

REFERENCES

[1] ISS Group, "About Us," in *ISS Group.com* [Online]. Available: http://www.issgroup.com.au/ABOUTUS/tabid/53/language/en-AU/Default.aspx [Accessed: August 21, 2008]

[2] ISS Group, "BabelFish™ - Connect, Interpret, Action" in *ISS Group.com.au* [Online], Available: http://www.issgroup.com.au/PRODUCTS/BabelFish/tabid/73/language/en-AU/Default.aspx [Accessed: October 13, 2008]

[3] D. Adams, *The Hitchhiker's Guide to the Galaxy*, London: Picador, 2005, p. 63

[4] ISS Group, "BabelFish™ Adapters" in *ISS Group.com.au* [Online], Available: http://www.issgroup.com.au/PRODUCTS/BabelFishComponents/BabelFishAdapters/tabid/74/language/en-AU/Default.aspx [Accessed: September 25, 2008]

[5] ISS Group, *BabelFish™ Installation Guide*. Perth, Western Australia : ISS Group, 2008

[6] A. Croft, R. Davison, and M. Hargreaves, *Engineering Mathematics: A Foundation for Electronic, Electrical, Communications and Systems Engineers*, 3rd ed. Essex: Pearson Education Limited, 2001, pp. 506-508

[7] ILI Technologies (2002) Corp, "Electric Submersible Pump," ilitech.com, 2005. [Online]. Available: http://www.ilitech.com/ElectricSubmersiblePump.html [Accessed: Sept. 8, 2008]

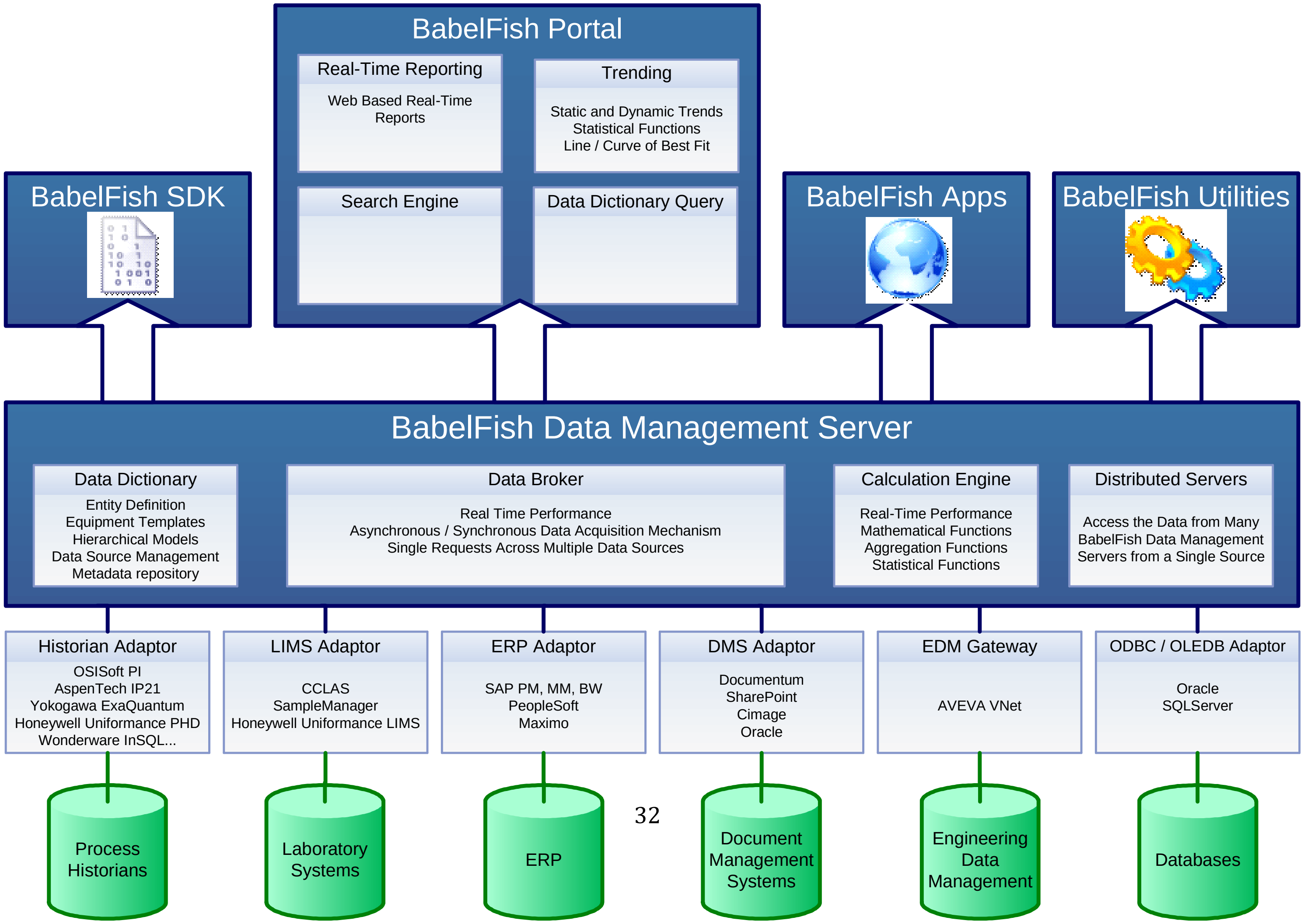
[8] RMS Enviro Solv Inc, "Godwin Heidra 200 Hydraulic Submersible Pump," rmsenviro.com, 2008. [Online]. Available: http://www.rmsenviro.com/godwin_hydraulic_submersible_Heidra_200.htm [Accessed: Sept. 10, 2008]

[9] Spinifex Group Pty Ltd, "Oil Water Tutorial," iprimus.com.au, 2008. [Online]. Available: http://home.iprimus.com.au/spiniwt/oil%20water%20tutorial.htm [Accessed: Sept. 11, 2008]

[10] ISS Group, *Detailed Design Document for the MED Operations Analysis; Corp, LNO and MKO Sites for Nickel West*; *Doc No. ISS-NIW-2007-108 v1.7 10/03/2008*. [Detailed Design Document]. Perth: ISS Group Australia Pty Ltd, 2008

[11] ISS Group, *Industrial Software Solutions: "Bridging the Gap between the process and IT"*. [Powerpoint presentation]. Perth: ISS Group Pty Ltd, 2004

APPENDIX A – BABELFISH™ FRAMEWORK [11]



APPENDIX B – INTERNSHIP TIMELINE

Current Date	12/11/2008
Title	Internship Plan
Project Start	4/08/2008 8:00:00 AM
Project Finish	21/11/2008 5:00:00 PM

ID		Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		BabelFish Training	16 wks	Mon 4/08/08	Fri 21/11/08		
2	✓	Murdoch - Project Planning	15 days	Mon 4/08/08	Fri 22/08/08		
3	✓	Murdoch - Progress Report	6 wks	Mon 25/08/08	Fri 3/10/08	2	
4		Murdoch - Final Repor	6 wks	Mon 6/10/08	Fri 14/11/08	3	
5	✓	Nickel West LNO Pag Building	3 days	Fri 8/08/08	Tue 12/08/08		
6	✓	BabelFish Portal Basi Training	2 days	Mon 4/08/08	Tue 5/08/08		
7	✓	BabelFish Portal Intermediate Training	2 days	Wed 6/08/08	Thu 7/08/08	6	
8	✓	BabelFish Portal Advanced Training	1 day	Fri 8/08/08	Fri 8/08/08	7	
9	✓	BabelFish Administrator Training	2 days	Thu 14/08/08	Fri 15/08/08	8	
10	✓	BabelFish Installation Training	2 days	Tue 19/08/08	Wed 20/08/08	9	
11	✓	BabelFish Basic Final Exercises	1 day	Tue 12/08/08	Tue 12/08/08		
12	✓	BabelFish Intermediat Final Exercises	1 day	Wed 13/08/08	Wed 13/08/08	11	
13	✓	BabelFish Admin Fina Exercises	8 days	Mon 18/08/08	Wed 27/08/08	12	
14	✓	ISS Wiki Updating	62 days	Thu 28/08/08	Fri 21/11/08		
15	✓	Nickel West KNS Tag Transfer	2 days	Thu 4/09/08	Fri 5/09/08		
16	✓	FMG Horizon 1 BabelFish Server Migration and Testing	6 days	Fri 26/09/08	Fri 3/10/08		
17	✓	Nickel West BabelFis 2.1.5.8 testing	5 days?	Tue 14/10/08	Mor 20/10/08		
18		Nickel West Horizon 2 LAU	15 days	Mon 3/11/08	Fri 21/11/08	17