



Arc Marine

GIS for a Blue Planet

Dawn J. Wright

Michael J. Blongewicz

Patrick N. Halpin

Joe Breman

Foreword by Jane Lubchenco

ESRI PRESS
REDLANDS, CALIFORNIA

ESRI Press, 380 New York Street, Redlands, California 92373-8100

Copyright © 2007 ESRI

All rights reserved. First edition 2007

10 09 08 07 1 2 3 4 5 6 7 8 9 10

Printed in the United States of America

Library of Congress Cataloging-in-Publication Data

Arc marine : GIS for a blue planet / Dawn J. Wright ... [et al.] ; foreword by Jane Lubchenco. -- 1st ed.
p. cm.

Includes bibliographical references.

ISBN 978-1-58948-017-9 (pbk. : alk. paper)

1. Oceanography--Geographic information systems. I. Wright, Dawn J., 1961--

GC38.5.A73 2008

551.460285--dc22

2007000708

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and the copyright laws of the given countries of origin and applicable international laws, treaties, and/or conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying or recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts and Legal Services Manager, ESRI, 380 New York Street, Redlands, California 92373-8100, USA.

The information contained in this document is subject to change without notice.

U.S. Government Restricted/Limited Rights: Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than restricted/limited rights. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, California 92373-8100, USA.

ESRI, ArcGIS, ArcCatalog, ArcMap, ArcGlobe, ArcSDE, ArcIMS, ArcScene, Geography Network, ArcWeb, ModelBuilder, and the ESRI Press logo are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

Ask for ESRI Press titles at your local bookstore or order by calling 1-800-447-9778. You can also shop online at www.esri.com/esripress. Outside the United States, contact your local ESRI distributor.

ESRI Press titles are distributed to the trade by the following:

In North America:

Ingram Publisher Services

Toll-free telephone: (800) 648-3104

Toll-free fax: (800) 838-1149

E-mail: customerservice@ingrampublisherservices.com

In the United Kingdom, Europe, and the Middle East:

Transatlantic Publishers Group Ltd.

Telephone: 44 20 7373 2515

Fax: 44 20 7244 1018

E-mail: richard@tpg ltd.co.uk

Cover and interior design by Savitri Brant



Contents

Foreword Professor Jane Lubchenco [vii]

Preface [ix]

Acknowledgments [xi]

Chapter 1 Introduction [1]

Chapter 2 Common Marine Data Types [9]

Chapter 3 Marine surveys [21]

Chapter 4 Marine animal data applications [45]

Chapter 5 Implementing time series and measurements [81]

Chapter 6 Nearshore and coastal/shoreline analysis [107]

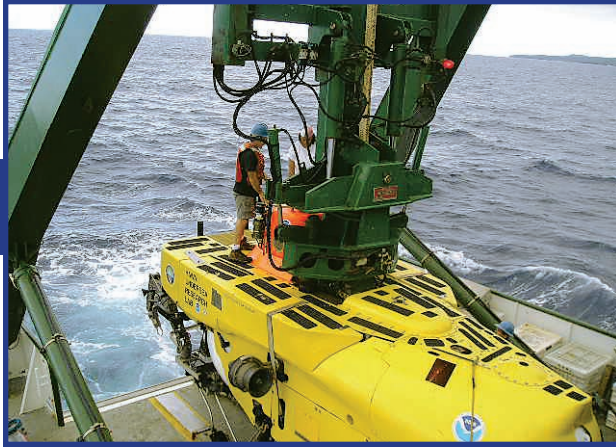
Chapter 7 Model meshes [141]

Chapter 8 Multidimensional GIS [163]

Chapter 9 Epilogue [177]

About the authors [183]

Index [185]

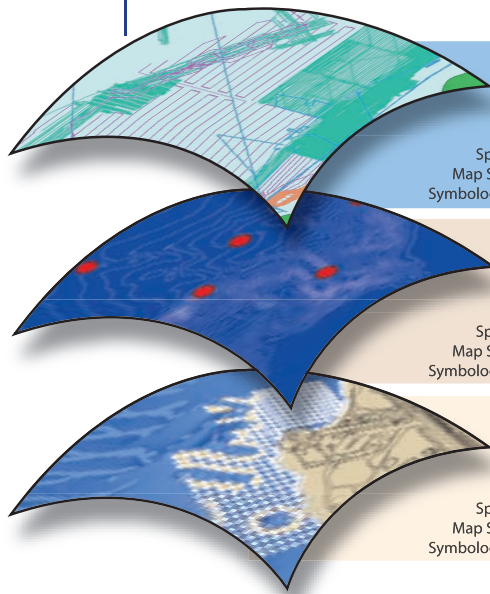


5

chapter five

Implementing time series and measurements

GIS has recently made great improvements in its ability to implement the temporal dimension (e.g., Langran 1992; Miller 2005). The improvements resulted partly from continued demands from user groups and partly from technological advances. This chapter describes how the Arc Marine data model sorts out the complexity of storing time-varying data, thereby providing users with a logical means of accessing the data for query, display, analysis, and map production.



Layer **Tracks and Cruises**
 Map Use Shiptracks during a cruise, tracks of vehicles towed from a ship or deployed from a ship untethered, autonomous
 Data Source Shipboard or vehicle GPS logs storing time, date, and position
 Representation Linear features
 Spatial Relationships Tracks have a direction with time stamps along route, particularly keep sampling stations
 Map Scale and Accuracy Typical map scales range from 1:24,000 to 1:50,000; locational accuracy ~10 m
 Symbology and Annotation Line symbology drawn with varying weights and patterns, annotated with date/time and ship/vehicle

Layer **TimeSeries Locations**
 Map Use Variations in time of variables measured at fixed observations stations at sea and onshore
 Data Source Fixed or moored measuring devices such as hydrophones, acoustic doppler current profilers (ADCP), ocean bottom seismometers (OBS), tide gauges
 Representation Point features
 Spatial Relationships Points can be related to center of a grid cell or associated with a time series or numerical model
 Map Scale and Accuracy Typical map scales range from 1:10,000 to 1:24,000; locational accuracy ~10 m
 Symbology and Annotation Point marker symbology with associated instrument attributes

Layer **Instantaneous Measured Points**
 Map Use Variations in space of variables measured at a given moment in time through the water column
 Data Source Instrument casts such as conductivity-temp-depth (CTD), expandable bathythermograph (XBT), sound velocity profile (SVP), fish density, etc.
 Representation Point features, vertical profiles
 Spatial Relationships Points can have varying depths associated with a single location, as well as multiple measurements
 Map Scale and Accuracy Typical map scales range from 1:10,000 to 1:24,000; locational accuracy ~10-50 m
 Symbology and Annotation Point marker and linear symbology annotated with associated instrument attributes

Introduction

Adding a temporal dimension to geographic data makes it more complex. However, users can better re-create the natural state of a feature and visualize the measured values at a given point in time once they have added a temporal perspective to the dataset.

For example, once they sort out the complexity, users could query a geodatabase for water temperature values during a given month of a specified year and then compare those values with the values of the same month in a different year. Additionally, users could track changes of a feature over time or the generation of statistics for a time period in order to compare those results against statistics generated for a separate time period. Furthermore, the symbology for those features could change as the values change from one point in time to another.

This chapter focuses on the element of time during the process of values being recorded at varying positions and varying depths. That element of time might be a fixed stamp, or it might be collected continually over longer periods, generating what is commonly referred to as a time series. Given that the dataset varies over time, the intent is to let users query and view their data at one period of time or another. In marine applications, this can include looking at the speed and direction of currents in order to understand sediment transport; discerning how daily and monthly fluctuations in sea surface temperature (SST) may correlate to phytoplankton distributions and fish migrations; and tracking the motion of an oil spill, a hydrothermal plume, or a warm core ring. All of these scenarios have an important element of time, and the data to support them is often based on measurements taken at varying depths.

Featured case study

The main case study for this chapter was implemented by researchers at the Marine Institute, Galway, Ireland, in collaboration with developers at ESRI Ireland, and is still ongoing. The marine resources of Ireland (220 million acres) constitute more than ten times its land area, resulting in a huge impact on the Irish economy currently in excess of 3 billion Euros in revenue and employing more than 44,000 people (Marine Institute 2005). The national Marine Institute of Ireland coordinates and promotes marine research, development, and conservation along with associated services (such as data integration, and access). All of this is provided with the goal of promoting economic development. The Marine Institute also collaborates on marine activities with a multitude of other government departments, agencies, industries, and research institutes in Ireland. As a result, it must deal with a plethora of datasets collected in a diverse range of formats and at varying standards and scales. Before it adopted Arc Marine, the Marine Institute stored and managed the majority of its datasets independently of each other, resulting in inadequate overview and management of national archives (Hennessy et al. 2006) in ways that included the following:

- An increased risk of data loss or corruption of files since they were not managed and controlled within a single database management system.
- Not exploiting the full scientific value of the data due to the inability to readily integrate data from multiple sources, due to the diverse current data storage arrangements.
- Responses to incoming requests for information involving a significant investment of time and effort, despite having the data readily available. For example,
 - What is the average summer temperature of Galway Bay?
 - How much have Irish waters warmed up over the last decade?
 - Is there any correlation between algal bloom events in Cork Harbor and variations in water temperature?

To address these issues, the Marine Institute developed a geodatabase called the Marine Data Repository (and the accompanying interface for querying the geodatabase called Map Viewer). The geodatabase was implemented with the Arc Marine structure and powered by Microsoft SQL Server 2000 and ArcSDE technology. The loading of data into the Marine Data Repository presented an initial challenge due to the large volumes of data involved (nearly 120 million data records) and the various transformations needed from the source formats. As a solution, the Marine Institute has used geoprocessing tools with Python scripts for the automatic loading of data. These ETL (Extract, Transform, and Load) scripts ran the SQL Server procedures necessary for the automatic import of the datasets and their transformations from source formats. Use of ETL scripts also keeps track of the various Arc Marine FeatureIDs and MeasurementIDs that are assigned, as these identifiers are vital for managing the various relationships within the geodatabase and ensuring the integrity of the data.

Figure 5.1 shows the extent of current holdings in the Marine Data Repository by geographic coverage and oceanographic subdiscipline. The initial implementation of Arc Marine focused on underway and conductivity-temperature-depth (CTD) data from one of the Marine Institute's research vessels (the R/V Celtic Explorer), international CTD data collected from Irish territorial waters, physical data from the Irish Marine Data buoy platforms,

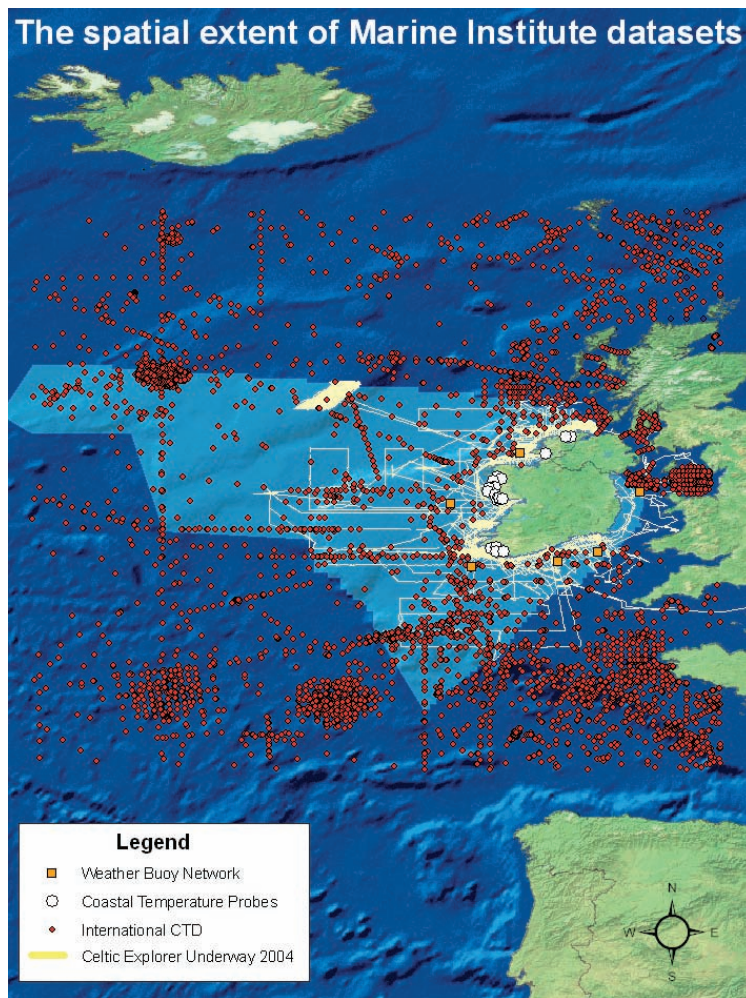


Figure 5.1 The extent of chemical and physical oceanographic data stored by the Marine Institute in the Marine Data Repository.

© Marine Institute, Ireland.

and nutrient monitoring and temperature data from coastal temperature probes, which were extremely time varying.

Time-varying data

The Arc Marine data model essentially has three structures for storing time-varying data. This chapter addresses two of those, while chapter 7 will cover the third. As presented in chapter 3, the InstantaneousPoint feature class is the best way to store time-varying data. This point feature class contains a time stamp as an attribute, and then through a series of relationship classes, the data values to be recorded at that location and time stamp

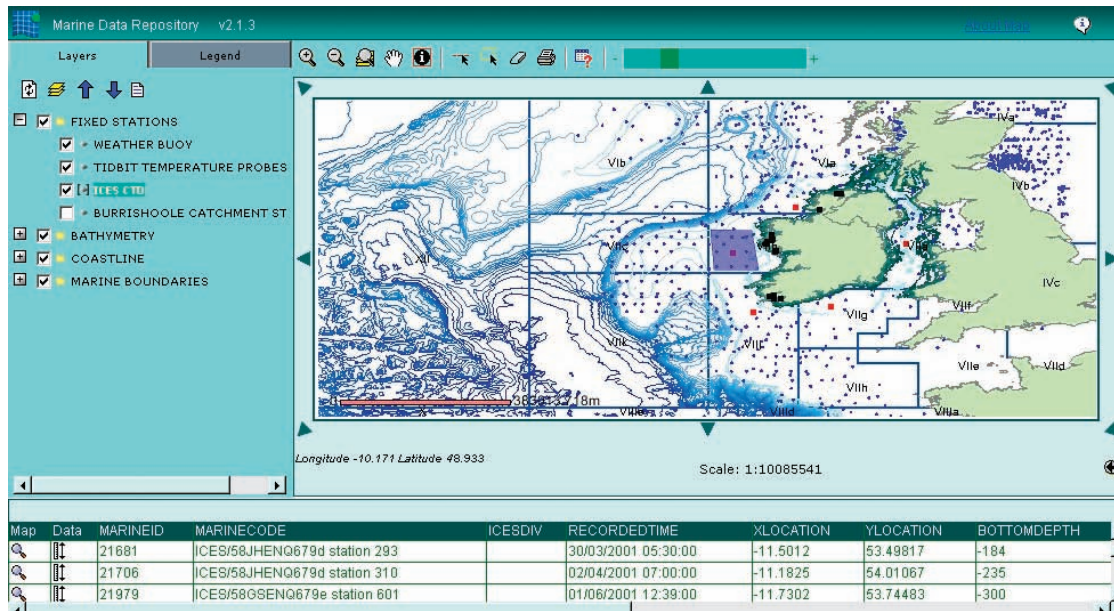


Figure 5.2 The results of a query for CTD sampling stations from the Marine Institute's Map Viewer Web GIS application (developed for the Marine Institute by ESRI Ireland and based on the Microsoft .NET framework).

© Marine Institute, Ireland.

are stored in an object class called `MeasuredData`. For any new measurement value being recorded, a new record is added to the `MeasuredData` table. The `MeasurementID` attribute is then carried over to identify where that value had been recorded. This structure can easily be associated with the storage of CTD data where multiple variables are being measured and collected at multiple depths for a single instant in time (figure 5.2).

A time series is the second way to store time-varying data. The time steps can be regular, meaning that the distance between one time step and the next is consistent throughout the time series, or it can be irregular, where values are measured at random intervals. The values can represent virtually any phenomenon that may change over time.

The Arc Marine data model lets users associate the time series with a spatial feature so that the spatial and temporal components can be combined in query, display, and analysis.

Measurement points

Introduced during the discussion of marine surveys in chapter 3, `MeasurementPoint` is an abstract class acting to thematically organize point feature classes that store features where measurements are being recorded. It has two subclasses that can be instantiated,

InstantaneousPoint and TimeSeriesPoint. The InstantaneousPoint feature class was also initially introduced in chapter 3, but the focus was on its use for storing survey data. Features of the InstantaneousPoint feature class are defined as being fixed in space and time. This means that a unique feature is defined by its x- and y-coordinate and a single time stamp. There are four subtypes available for this feature class, Instant, Sounding, Survey, and LocationSeries, with Instant being the default. Although the subtypes are treated and act the same, this chapter will focus on the use of the Instant subtype. Refer to chapter 3 for the complete description of this data structure.

The TimeSeriesPoint feature class, like the InstantaneousPoint feature class, is a subclass of MeasurementPoint. This feature class introduces no new attributes and is designed to be a general feature class for the variety of features established for collecting data over long periods of time. Features of TimeSeriesPoint are generally associated with a time series, whereas the use of InstantaneousPoint is associated with a single time stamp. Although Arc Marine does not limit the association of time series to any feature class, regardless of its geometry, the TimeSeriesPoint feature class is offered in this data model as a template. For example, a moored buoy might have several measuring devices attached for measuring values of wave height, sea surface temperature, wind speed, and direction. Where the moored buoy would be represented as a feature of the TimeSeriesPoint feature class, the values for each of these variables being recorded over a long period of time would be stored as an individual time series. Arc Marine then provides the framework in which these time series can be associated with the correct buoy feature.

Features of TimeSeriesPoint or InstantaneousPoint can have multiple depths. In the previous example of the moored buoy, additional measuring devices might be cabled together at varying depths below the sea surface to record additional data such as salinity, water column temperature, current speed, and direction. These would be stored as a time series and associated with the moored buoy feature at their recorded depth. As with InstantaneousPoint, users accomplish this through the Measurement object class. A section later in this chapter titled Measurements details how users manage this within Arc Marine.

In the case study developed by the Marine Institute of Ireland, the InstantaneousPoint and TimeSeriesPoint feature classes are the only feature classes of Arc Marine that are implemented. However, the Institute has extended a number of object classes extensively to accommodate its applications and to make the collected data available to all staff members (figure 5.4). The additional tables have been grouped into either “Activities” (an organizational mechanism based on the collection activity), “Organizations,” “Contacts,” or “Platforms,” all of which describe who owns the data, who to contact about accessing data, and where a data collection is being performed. The Marine Institute uses InstantaneousPoint to store its research vessel underway data, as well as datasets such as CTD casts collected from the vessel. TimeSeriesPoint is used for storing its offshore weather buoy, nutrient monitoring, and coastal temperature probe locations. Both the InstantaneousPoint and TimeSeriesPoint feature classes have an ActivityID attribute added for relating the features to the Activity table (figure 5.4). In this way, the institute is able to make its datasets more readily available and to derive value-added products such as monthly climatologies (hence information, and data).

Common Marine Data Types

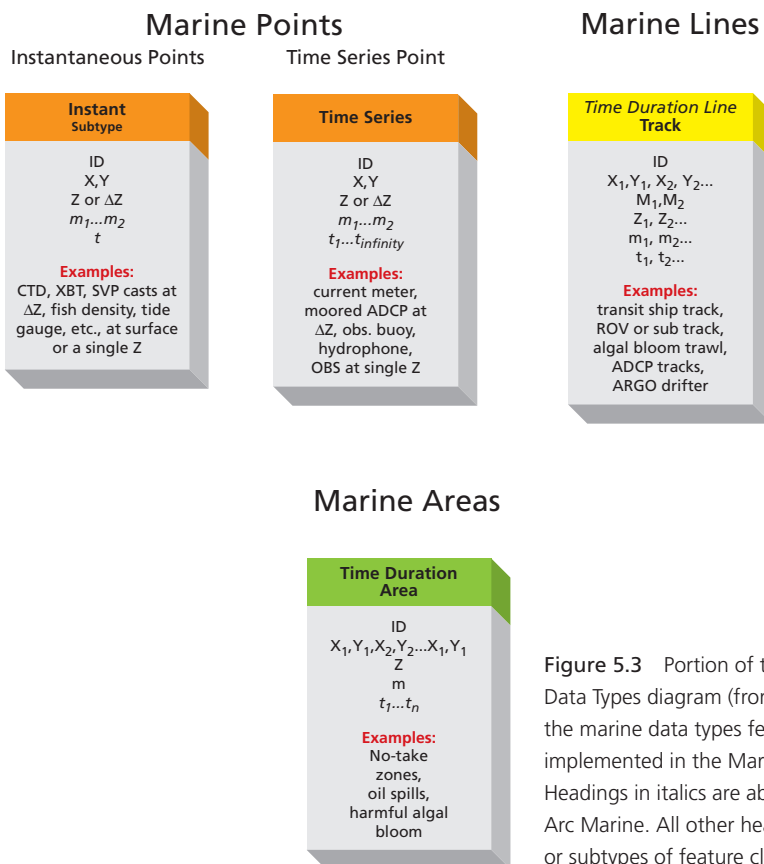


Figure 5.3 Portion of the main Common Marine Data Types diagram (from chapter 2) representing the marine data types featured in this chapter and implemented in the Marine Institute case study. Headings in italics are abstract feature classes in Arc Marine. All other headings are feature classes or subtypes of feature classes.

TimeSeries

The data structure for the TimeSeries classes implemented in Arc Marine is the same data structure introduced and implemented in the Arc Hydro data model (Arctur and Zeiler 2004; Maidment et al. 2002). Using data over time is just one aspect in common between the Arc Hydro and Arc Marine data models, and for the sake of constancy it was important to this effort that a new data structure for dealing with time-varying data not be introduced. The TimeSeries group consists of two object classes and one relationship class. The first table, TimeSeries, is the table in which the data values are actually stored. This table consists of four fields, FeatureID, TSTypeID, TimeValue, and DataValue. The value of FeatureID identifies the feature associated with the time series and is used as a key field for developing any relationships between this table and a feature class. The TSTypeID is a key field for the relationship class TSTypeHasTimeSeries between the TimeSeries table and the

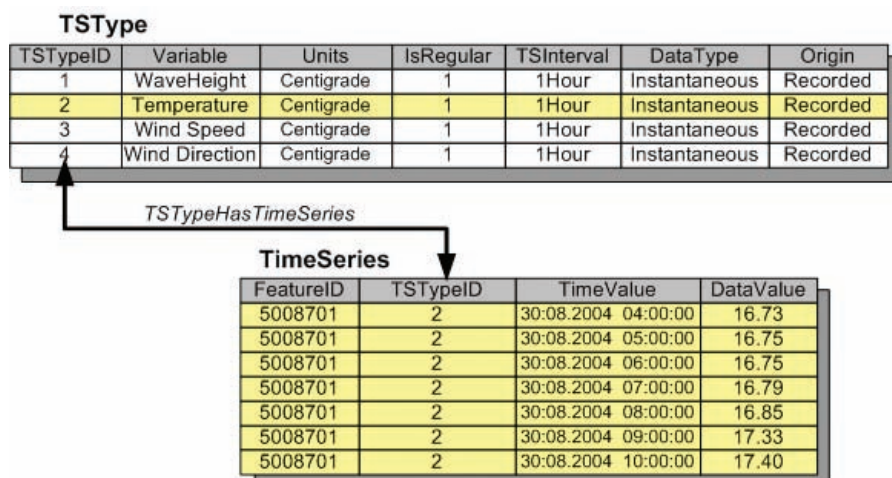


Figure 5.5 The structure of the TSType table, the TimeSeries table, and the relationship between them.

TSType table. It identifies the record in the TSType table, which defines the parameters of the time series. TSDateTime is the field (and is of a Date type that includes time) for storing the time steps. TSValue is the field storing the corresponding data values. These last two columns essentially make the time series matrix.

The TSType table is used for storing parameters that describe the time series data (figure 5.5). TSType acts as a lookup table for all data values stored in the TimeSeries table. Consequently, the TSTypeID is meant to be a unique identifier for each “type” of time series in a geodatabase and acts as a key field for relating to the TimeSeries table. How a user defines a type of time series should be flexible. The TimeSeries table provides several options. The Variable field is a string field that identifies the measured data. Examples of this include wave height, temperature, and salinity. The Units field is a string field that stores information with the units of measure for the collected data. The IsRegular field is a Boolean field identifying whether the time step interval is regular or irregular. TSInterval defines the time interval between two time steps of a Regular time series. For irregular time series, TSInterval is undefined. The values of TSInterval are defined in a coded value domain called TSIntervalType and include values such as 1Minute, 2Minute through 30Minutes, 1Hour, 2Hour, 1Day, and 1Month. The DataType field determines how the type of stored data has been recorded and uses a coded value domain that includes Instantaneous, Cumulative, Incremental, Average, Maximum, and Minimum data. The Origin field uses a coded valued domain of either Recorded or Generated. A Generated origin generally refers to a time series created from a model, whereas a Recorded time series consists of actual measured values.

There is no defined relationship class between the TimeSeriesPoint feature class and the TimeSeries table. This presumes that when users implement the TimeSeries feature class, it would be converted to an abstract class and inherited by a user-specified subclass. For example, users might build specific feature classes for the points representing weather buoys and

another feature class for temperature probes. The two subclasses of `TimeSeriesPoint` would inherit from that class. Through this process, the `FeatureID` is present in each of the feature classes and in the `TimeSeries` table. With this attribute, a relationship is implied.

Accessing and applying time series data

Accessing and using time series data stored in a geodatabase is the same as accessing any other table in the geodatabase. Users can query the set of tables and graph the results. Depending on the question, users can query the table based on a combination of the ID of the feature, time period, and type of data, essentially providing the capability to query for a particular variable in space and time. For example, users might query the `TSType` table for a specific type of data, such as temperature. Then the selection set of the `TimeSeries` table is updated via the relationship class `TSTypeHasTimeSeries`. This results in a selection set of the `TimeSeries` table for the values representing temperature data. Based on the `FeatureID` or a specific time period, and using values within a specified range, users can expand the query by selecting from the current selection set of records. This results in isolating the identifiers of the features where temperatures for a given time period and of a particular value or value range are associated.

However, going beyond this query example usually requires some additional processing. For example, to highlight features in the ArcMap data view associated with these time series, users need to build an additional relationship between the feature class and the `TimeSeries` table, using `FeatureID` as the key field. As described above, the `FeatureID` exists in both the feature class and `TimeSeries` table. Once this relate has been built, the associated features can be highlighted. However, a relate doesn't allow for the use of the `TimeSeries` values for rendering in ArcMap. Additionally, a Join in ArcMap doesn't support a one-to-many relationship between a feature and all of the records included in a time series. Users could generate Summary Statistics for the selected set of the `TimeSeries` table as the input. This produces statistics for each unique `FeatureID` and `TSType` combination. The output table from that process could then be joined to the feature class, which then could be used to update the symbology of that feature class. Using the example of the moored buoy, we might presume that a user has several instances of these features, which have been collecting temperature data for the past year. Once that data has been imported into the `TimeSeries` table, the user could go through a similar process outlined above and update the display categorically for those `TimeSeries` points based on the statistical results.

Advanced applications for managing time series data are starting to be developed and made available as extensions to ArcMap. For example, DHI Water & Environment, an international consulting and research organization, has recently released an extension to ArcMap called the Temporal Analyst for ArcGIS (see the book's companion Web sites for more details). With this extension, users can easily import time series data into a geodatabase from a variety of sources, associate the time series with features, and then access the time series data for various plotting, reporting, and analytical processes (figure 5.6). The Temporal Analyst extension provides this functionality on a collection of time series data, individual

time series, or by directly accessing the time series data from the ArcMap data frame by right-clicking the feature and selecting the time series of interest (figure 5.7). The DHI Temporal Analyst extension also provides an analysis environment in which several advanced temporal processes are available, such as filling gaps in the data, resampling, synchronization, creating distribution curves, scatter curves, and double mass curves.

More advanced spatiotemporal analysis, through the use of tools such as statistical calculations and a temporal calculator, are also available through the Temporal Analyst extension. Within the analysis environment, the example outlined above is a matter of a few mouse clicks. Once the time series of interest are added to the analysis environment of the Temporal Analyst extension, they can be selected and the Compute Scalar(s) From Time Series dialog box can be opened (figure 5.8). In the Compute Scalar(s) From Time

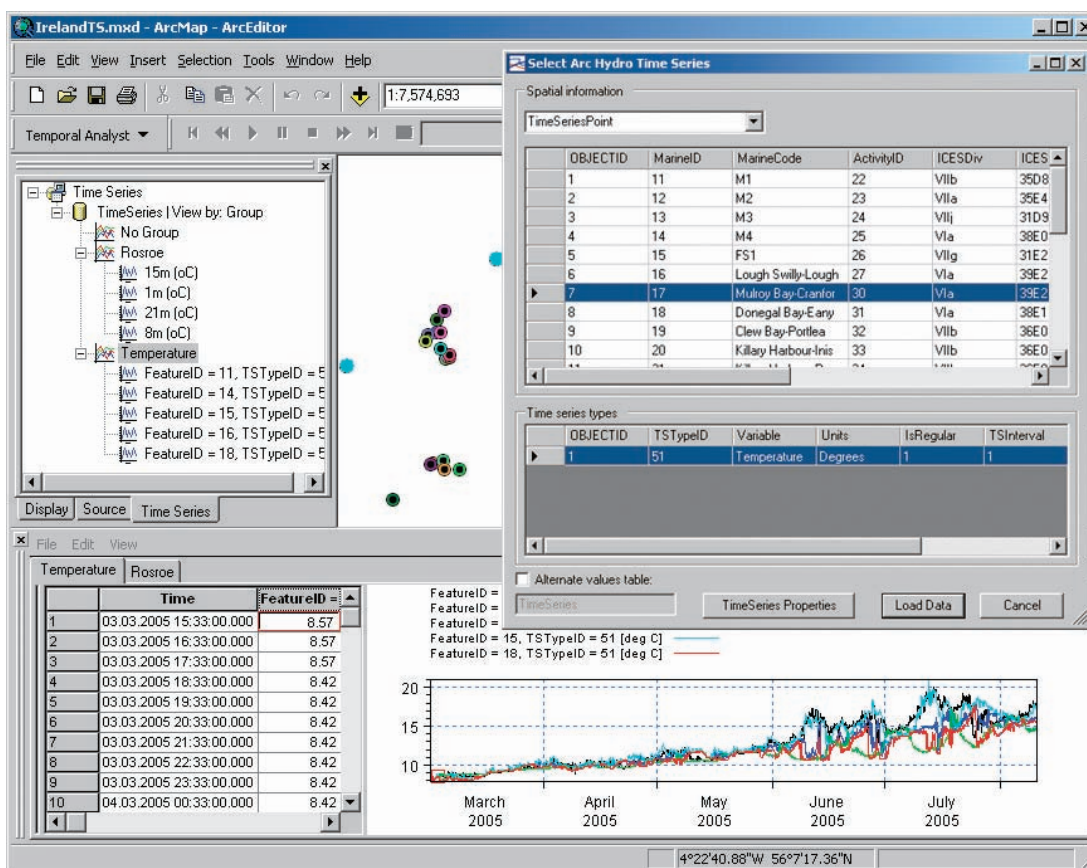


Figure 5.6 The DHI Temporal Analyst extension provides access to Temperature data stored in an Arc Marine geodatabase. The Marine Institute provided this geodatabase. As an example, the geodatabase shows temperature data associated with temperature probes stored in the TimeSeriesPoint feature class.

© Marine Institute, Ireland.

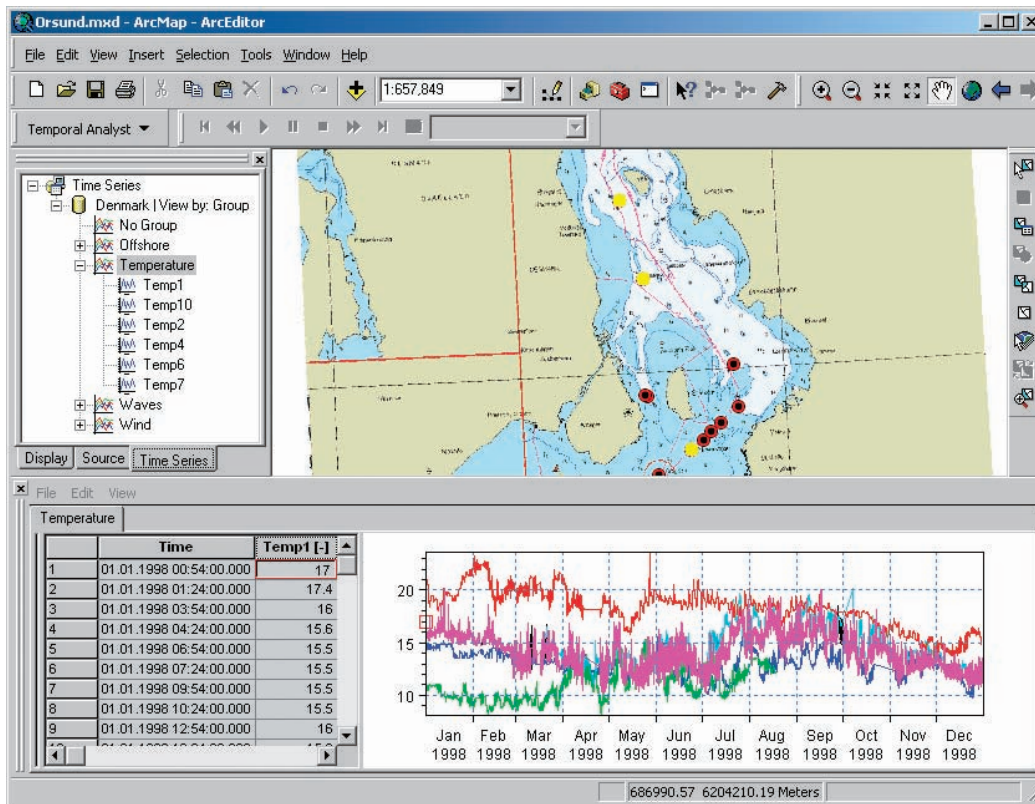


Figure 5.7 The DHI Temporal Analyst extension as used in ArcMap to plot a collection of temperature time series data collected in the Kattegat Strait between Denmark and Sweden and stored in the geodatabase.

Series dialog box, the time period of interest is specified, and the Time Weighted Average is selected for the statistical operation. This operation produces a feature class that is a replica of the features associated with the input time series and with the statistical value added to the attribute table. This attribute can then be used to determine the categorical distribution and symbology for the new feature layer, as seen in figure 5.8. The new feature layer, in this case a point feature layer, can alternatively be used as input for interpolating a statistical surface using the inverse distance weighted (IDW) function of the ArcGIS Spatial Analyst extension (figure 5.9).

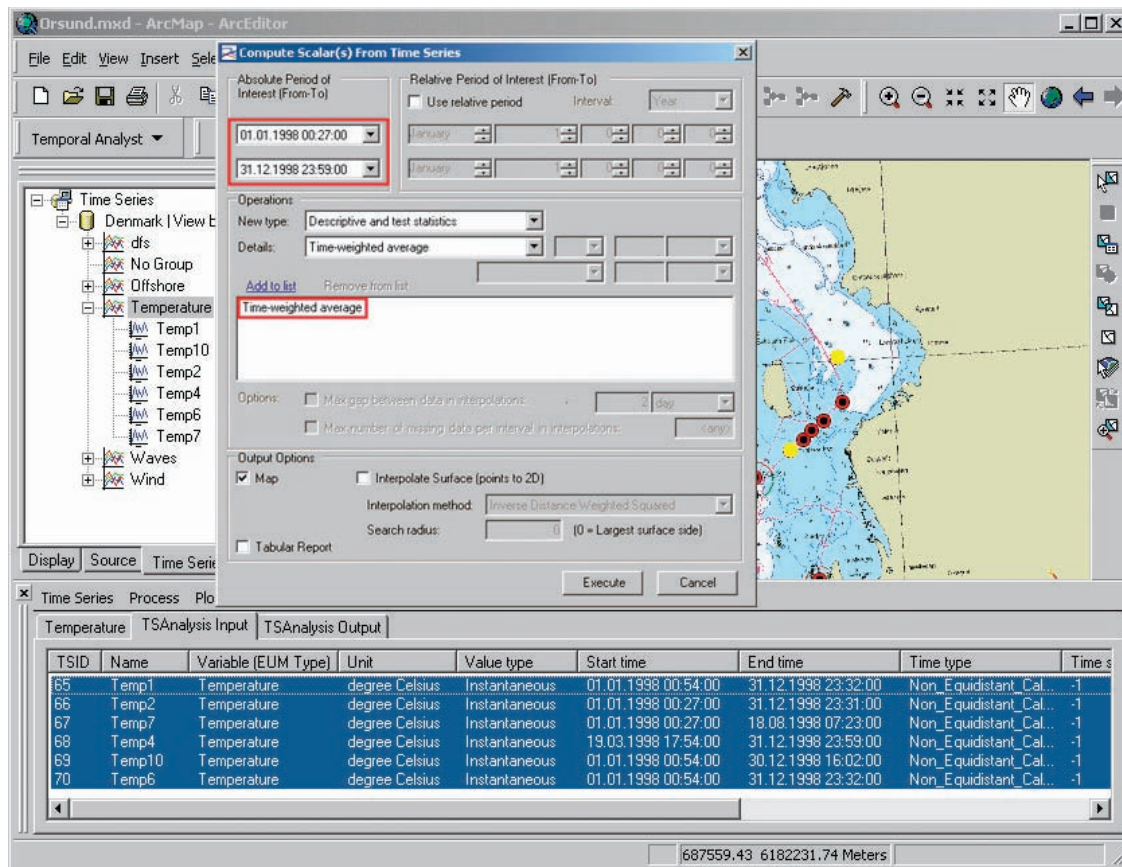


Figure 5.8 The Compute Scalar(s) From Time Series function of the DHI Temporal Analyst extension calculates the time-weighted average of the selected time series.

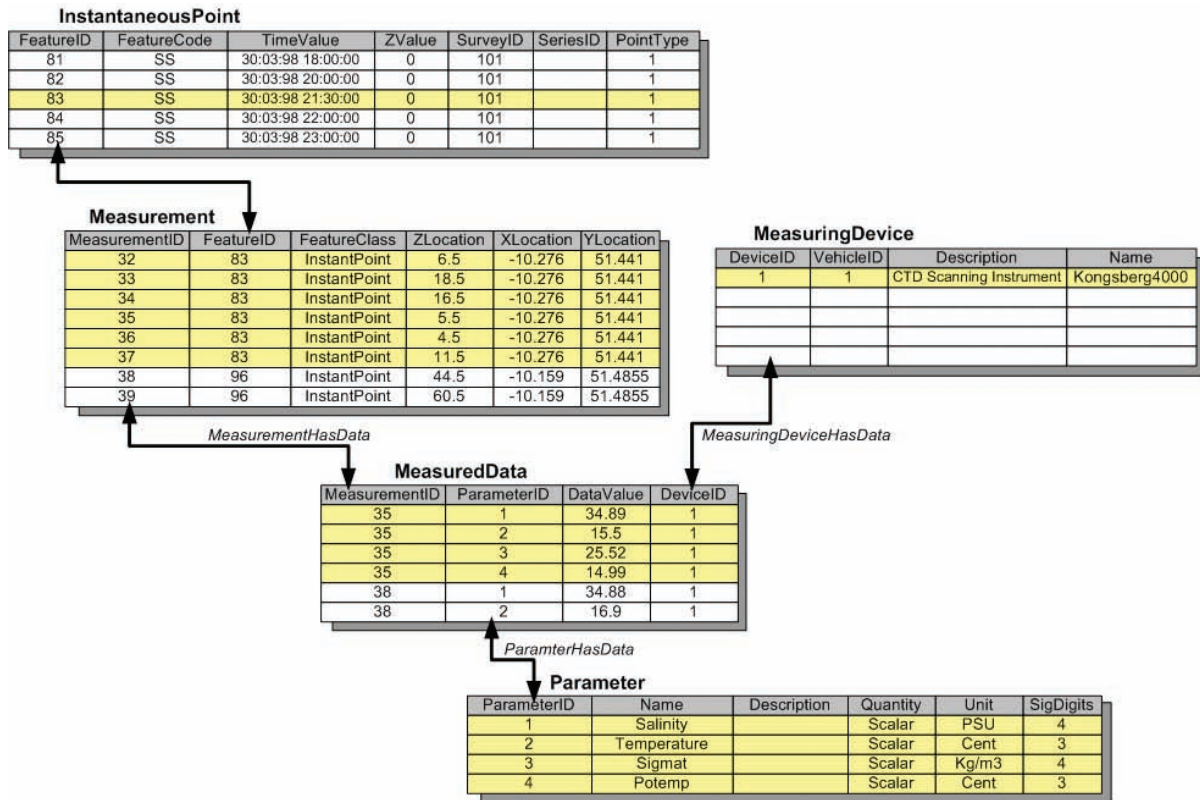


Figure 5.9 The results of computing the Time Weighted Average of the Temperature time series and associating that scalar value to the point features associated with the time series. The scalar value is used to determine the symbology for the measuring stations and further used as the input value for interpolating an IDW surface using ArcGIS Spatial Analyst.

Limitations to the TimeSeries classes

The Arc Marine development team understood the importance of integrating this data model with the data models of similar applications. Since a data structure for time series had already been published with the Arc Hydro data model, it was natural that Arc Marine should implement the same data structure. However, Arc Hydro team members did not have a lot of experience combining spatial and temporal data when they were developing the data structure for time series. It was important to keep the data structure simple with the ability to resolve simple queries for locating a feature with a given variable or type of time series and for a specified time period. Arc Marine meets that requirement. However, since the development of Arc Hydro, and with the advent of other technologies, the use of temporal and spatial data has increased beyond query and graphing. We have more

ERRATUM: Figure 5.9 on page 94 is incorrect. Here is the correct Figure 5.9. The caption in the book is correct as printed.

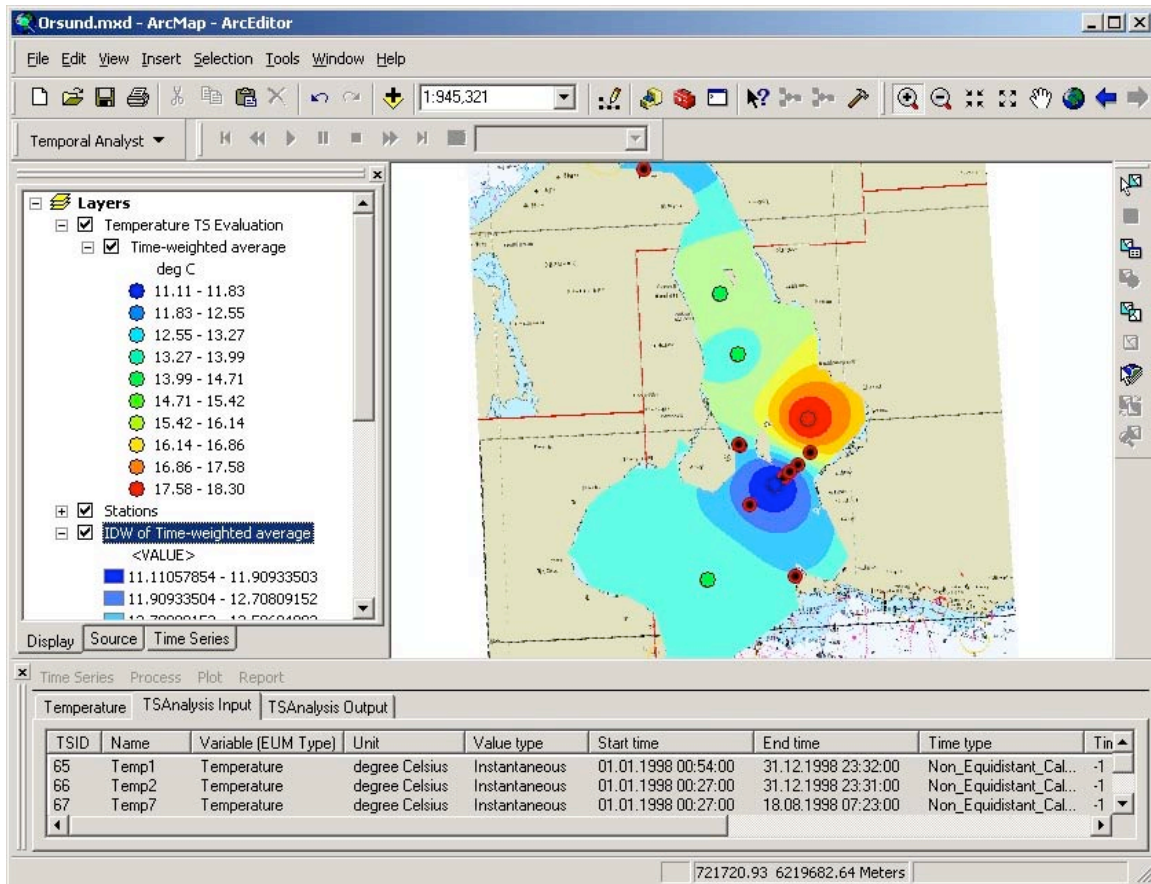


Figure 5.9. The results of computing the Time Weighted Average of the Temperature time series and associating that scalar value to the point features associated with the time series. The scalar value is used to determine the symbology for the measuring stations and further used as the input value for interpolating an IDW surface using ArcGIS Spatial Analyst.

experience and knowledge about the best ways to manage temporal data, and advances have exposed some limitations to the original time series data structure.

By having the FeatureID attribute included in the TimeSeries table along with the TSTypeID attribute, a feature can only be associated with a single instance of a time series of a given type. For example, stream flow can be associated with a gauging station. Generally, a gauging station will be associated with a single stream flow time series. This may be sufficient for mere data collection and input. However, in other instances, the time series might represent a forecast scenario, a modeling result, or a user might have several versions based on a particular event or different time periods. In those cases, the type of time series replicated for each instance of that feature would be identical and would also need to be associated with multiple versions of a similar time series.

Because a time series is a matrix, the FeatureID is repeated for every time step of a time series in the TimeSeries table. This precludes having a specific time series associated with more than one feature. That can be resolved by duplicating the time series matrix and populating the FeatureID for each record with a new value; but that is not good database management and not recommended.

To solve these limitations, first add an additional field to the TimeSeries table, called TimeSeriesID. This field distinguishes one time series (a collection of rows in the TimeSeries table) from another, and is populated with the same value for each row being included. Second, add a simple table that has just three attributes, TimeSeriesID, FeatureID, and FeatureClass. This table acts as a turntable or attributed relationship class, in which the TimeSeriesID and FeatureID are key fields through which two relationship classes would link to the TimeSeries table and to the feature class, and match up with the TimeSeriesID and FeatureID, respectively. In the turntable, the TimeSeriesID and FeatureID are populated with any number of combinations, and any combination can be repeated any number of times. The FeatureClass attribute is not required, but aids in identifying the feature class where the FeatureID is located.

Measurements

As previously illustrated in chapter 3, features can have multiple depths associated with them, where various variables are being recorded or measured. These measuring locations below the surface can be modeled in Arc Marine with the use of the Measurement table or what has been termed Measurements. The Measurement table provides the capability of having multiple depth locations for a specific feature. It has a FeatureID attribute designed to be used in a one-to-many relationship, between itself and the TimeSeriesPoint or InstantaneousPoint feature classes, for example. This table is where one point feature can have many entries, each describing a unique depth. The Measurement table also has a MeasurementID designed as a unique identifier for each Measurement in the geodatabase. Every combination of FeatureID and ZLocation creates a unique MeasurementID. For this reason, it is imperative to the integrity of the data that MeasurementID remains unique throughout the entire geodatabase.

When associating time series to a depth location for a given feature via the Measurement table, the value of the MeasurementID needs to be inserted as the value of the FeatureID in the TimeSeries table. Measurement acts as a virtual feature, which can be treated in a similar manner to physical features, while adding yet another dimension. In the temperature query example previously outlined, a query was made for temperature data, and the corresponding features were identified. Now, the depths of those features can be identified rather than simply identifying the associated features. Referring back to the DHI Temporal Analyst extension, if the Measurement table is populated, then making an association between a time series and a Measurement is supported. Likewise, as illustrated in figure 5.10, the time series associated with a Measurement can be retrieved.

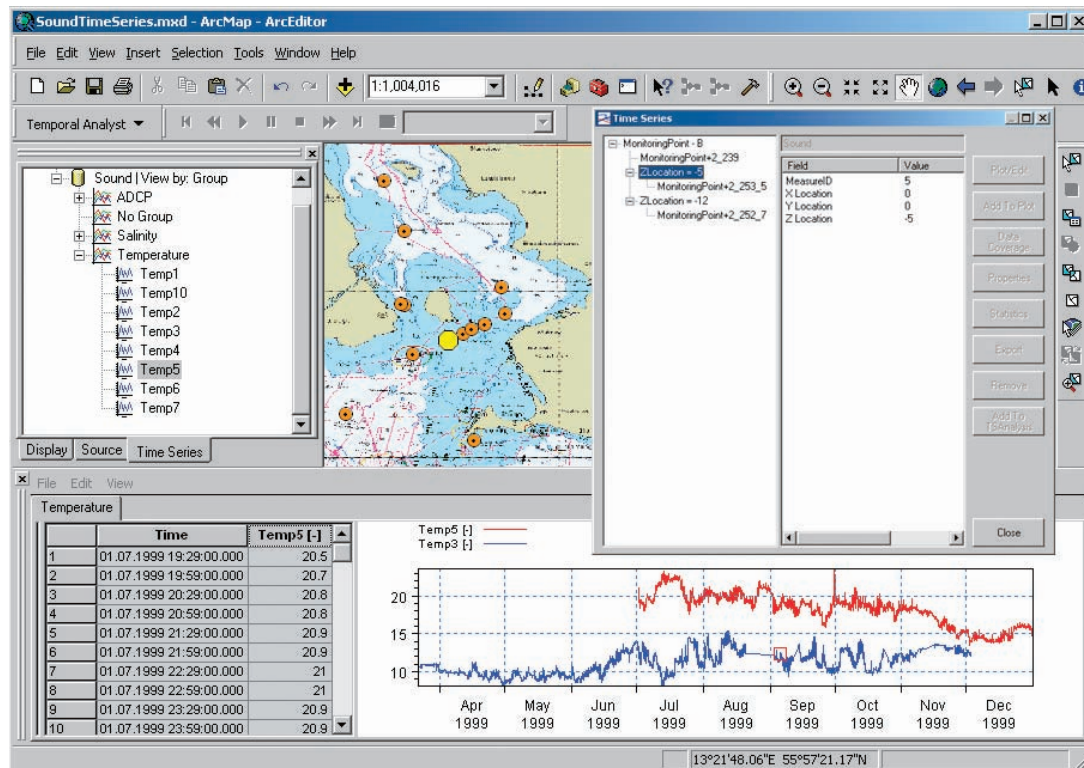


Figure 5.10 The DHI Temporal Analyst extension allows users to access and view the time series associated with the Measurements of a TimeSeriesPoint.

MeasuredData

The Arc Marine development team designed the MeasuredData table as an alternative to time series tables to store recorded or measured values for multiple variables. The generic table included in Arc Marine is available to store undefined values. The Parameter table, which defines these variables, can be accessed from the MeasuredData table via the ParameterHasData relationship class (figure 5.11).

The MeasuredData table has a MeasurementID attribute, which is the key field used in the relationship class MeasurementHasData. The relationship links the MeasuredData table with the Measurement table. This mechanism associates measured or recorded values with the depth of a point feature. The MeasurementID is repeated in this table for each variable measured at a specific depth.

For example, using this table to store the data from a CTD cast, the location of the cast would be represented as an InstantaneousPoint (denoted as x,y coordinates plus a time step). The various depths where variables were being recorded would be stored as Measurements in the Measurement table. The MeasuredData table would include an entry for each of the recorded variables for each Measurement. Figure 5.11 shows an example that includes entries for Salinity, Temperature, Sigma-t (water density), and Potemp (potential temperature) at each Measurement. Only the values appear in the table. The ParameterID is a key

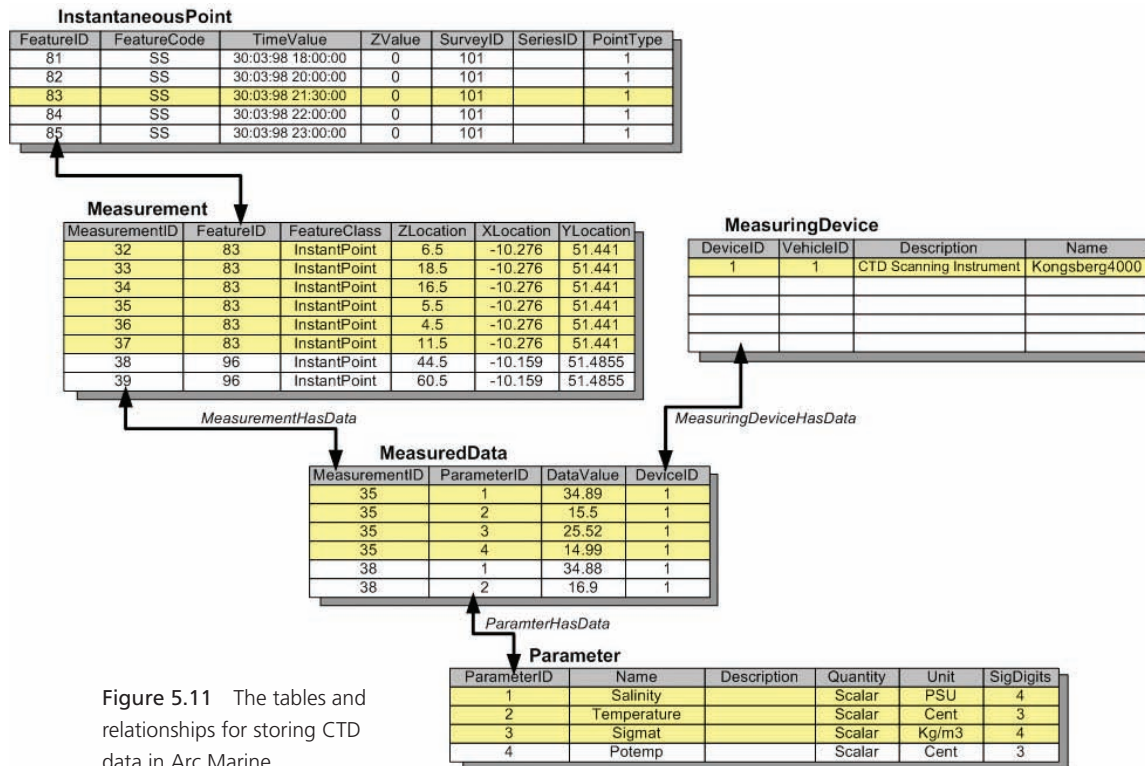


Figure 5.11 The tables and relationships for storing CTD data in Arc Marine.

field in the relationship class ParameterHasData, which connects this table to the Parameter table to identify the collected data type. This process is more efficient and flexible than adding an attribute to the MeasuredData table for each measured variable in the geodatabase.

The Marine Institute implements this structure for managing and archiving its CTD data. The institute designed a Web-based Map Viewer (figure 5.2) to make these datasets readily and easily available to its user base internal and external to the institute and in response to requests from other government agencies. The Map Viewer includes a Query Tool for easier access to the institute's complex storage structure (figure 5.12). The Query tool allows users to search the geodatabase using a number of different criteria:

Location	User-defined areas or manually entered coordinates
Time	Date ranges or recurring time periods, for example, every January
Depth	Either single values or depth ranges
Parameter	Select from one or more parameter groups

Advanced search criteria include the following:

Parameters	Select individual parameters associated with a group
Data Sources	Matches this list to Activities, allowing users to restrict the search to datasets of interest
Filter	Search for user-defined values or ranges of values

Figure 5.12 The Query tool within the Marine Institute's Map Viewer. The complex functionality facilitates searches on location, time, depth, and individual parameters. Additional filters on individual parameter values can be applied to focus the search.

Figure 5.13 shows the Vertical Measure feature within the Marine Institute's Map Viewer. For a given feature, this tool can display the results of a Join between the Measurement table and matching values of the MeasuredData table. The result is a listing of all measured CTD values at each depth.

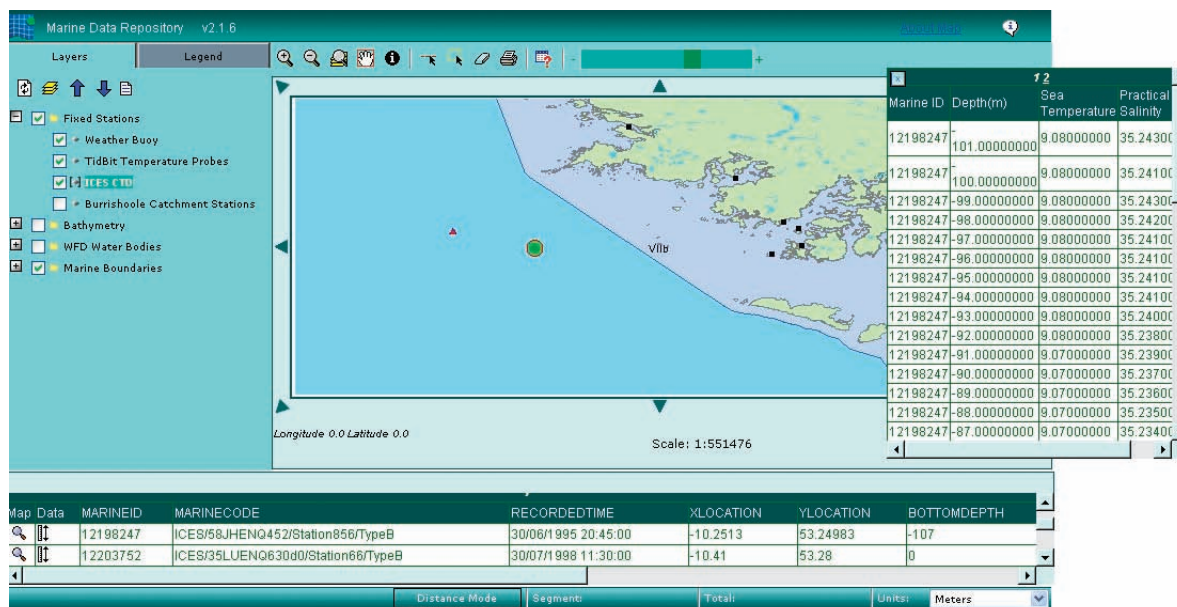


Figure 5.13 Result of a query using the Vertical Measure feature within the Query tool of the Marine Institute's Map Viewer, showing a listing of measured CTD values at varying depths, suitable for constructing a vertical profile. The CTD records are similar to the display in figure 5.2.

© Marine Institute, Ireland.

Conclusion

This chapter focused on the element of time, as short-term time stamps or longer-term time series, in conjunction with oceanographic measurements made at varying positions and depths. The case study of the Marine Institute of Ireland, through its Marine Data Repository, showed several examples of how Arc Marine can be effectively implemented with a variety of time- and space-varying measurements such as CTD casts or data collected from buoys or probes. It was further shown that with an extension such as the DHI Temporal Analyst, these point features can be associated with time series for plotting, querying, summary statistics, resampling, interpolating, weighted averaging, and more. These examples show that after sorting out the complexities of multidimensional data, the data model actually facilitates the building of applications that simplify access and allow for advanced processing. With the initial success of integrating physical and chemical oceanographic data into the Marine Data Repository, the inclusion of fisheries, biotoxin, and other

biological datasets is planned for the near future. These datasets will likely be incorporated into additional Arc Marine feature classes such as TimeDurationLines for fisheries hauls and TimeDurationAreas for harmful algal blooms.

The Marine Institute is still in the initial stages of its implementation of Arc Marine, with the maintenance of current data and the input of new data along with efforts to expand the geographic coverage for future holdings. Once the Marine Data Repository has been operational for a period of time, it is anticipated that enhancements and modifications will become clear based on user feedback. Meanwhile, the work represents a significant step forward for the Marine Institute in its management of data assets and delivery of improved information services. According to Hennessy et al. (2006), the implementation of Arc Marine has done the following:

- Provided a design template that would otherwise have been much more time-consuming to create and taken considerably longer to implement
- Forced issues such as varying data quality and standards to be addressed and resolved
- Put in place a single, secure, centralized, spatially enabled database on one server, thereby reducing the support requirements compared with a host of databases and formats previously used
- Allowed the integration of diverse marine datasets, facilitating a more holistic approach to the exploration and analysis of data
- Extended the services available from the institute to the wider scientific community, facilitating greater access to and dissemination of data and, over time, information
- Facilitated the future expansion of its management system regarding loaded data and software tools available to enhance analysis

Arc Marine class definitions featured in this chapter

FEATURE CLASSES	InstantaneousPoint —A feature class representing features that are single observations in time and space. The x- and y-coordinates, plus a time stamp, create the unique feature. An InstantaneousPoint can have multiple ZValues by implementing a relationship to the Measurement object class.		
	Subtype	Survey	
	Notes	InstantaneousPoint is a subclass of the superclass MeasurementPoint.	
	Properties	None	
	Fields	FeatureID	A geodatabase-wide unique identifier and key field for participating in relationships
		FeatureCode	A user-defined code used for identifying a feature
		CruiseID	A key field for relating this feature class to a Cruise
		TimeValue	The time stamp for the point
		ZValue	A single depth value for the point
		SurveyID	A foreign key to the SurveyInfo object class
		SeriesID	A key field for relating this feature to the Series table
		PointType	Defines the subtype to be one of the following: 1 = Instant (default value) 2 = Sounding 3 = Survey 4 = LocationSeries
	TimeSeriesPoint —A feature class representing features that are fixed in space (x,y) and have an association with an instance of the TimeSeries.		
	Subtype	None apply	
	Properties	None apply	
	Notes	A TimeSeriesPoint is a subclass of the superclass MeasurementPoint and can have multiple ZLocations by implementing the Measurement object class.	
	Fields	FeatureID	A geodatabase-wide unique identifier and key field for participating in relationships
		FeatureCode	A user-defined code used for identifying a feature
		CruiseID	A key field for relating this feature to the Cruise table
	ProfileLine —A feature class representing linear features that are not physical features themselves but rather features interpolated along the line from another source. For example, a profile interpolated from a bathymetry survey.		
Subtype	None apply		
Properties	HasM = True HasZ = True		
Notes	None available		
Fields	FeatureID	A geodatabase-wide unique identifier and key field for participating in relationships	
	FeatureCode	A user-defined code used for identifying a feature	

FEATURE CLASSES (cont'd)	FeatureLine —A feature class representing fixed or dynamic linear features.													
	Subtype	None apply												
	Properties	None available												
	Notes	None available												
	Fields	<table border="1"> <tbody> <tr> <td>FeatureID</td> <td>A geodatabase-wide unique identifier and key field for participating in relationships</td> </tr> <tr> <td>FeatureCode</td> <td>A user-defined code used for identifying a feature</td> </tr> </tbody> </table>	FeatureID	A geodatabase-wide unique identifier and key field for participating in relationships	FeatureCode	A user-defined code used for identifying a feature								
FeatureID	A geodatabase-wide unique identifier and key field for participating in relationships													
FeatureCode	A user-defined code used for identifying a feature													
OBJECT CLASSES	SurveyInfo —Designed for storing information about a specific survey.													
	Notes													
	Fields	<table border="1"> <tbody> <tr> <td>SurveyID</td> <td>A key field for relating this table to a feature class</td> </tr> <tr> <td>StartDate</td> <td>The beginning date of the survey</td> </tr> <tr> <td>EndDate</td> <td>The ending date of the survey</td> </tr> <tr> <td>Description</td> <td>A general description of the survey</td> </tr> <tr> <td>DeviceID</td> <td>A key field for relating a survey with a Measuring device</td> </tr> <tr> <td>TrackID</td> <td>A key field for relating a survey with a Track</td> </tr> </tbody> </table>	SurveyID	A key field for relating this table to a feature class	StartDate	The beginning date of the survey	EndDate	The ending date of the survey	Description	A general description of the survey	DeviceID	A key field for relating a survey with a Measuring device	TrackID	A key field for relating a survey with a Track
	SurveyID	A key field for relating this table to a feature class												
	StartDate	The beginning date of the survey												
	EndDate	The ending date of the survey												
	Description	A general description of the survey												
	DeviceID	A key field for relating a survey with a Measuring device												
	TrackID	A key field for relating a survey with a Track												
	Measurement —Designed for extending a single feature with multiple depths.													
	Notes	Measurements are designed for storing multiple depths for a single feature. This is done by creating a new and unique identifier for the feature, in this case MeasurementID, for every unique combination of FeatureID and ZLocation. In this table, FeatureID is repeated for every depth (ZLocation) associated with the feature.												
	Fields	<table border="1"> <tbody> <tr> <td>MeasurementID</td> <td>A unique identifier for the feature created by combining unique combinations of FeatureID and ZLocation</td> </tr> <tr> <td>FeatureID</td> <td>A key field relating to the FeatureID of the feature</td> </tr> <tr> <td>FeatureClass</td> <td>The name of the feature class in which the relating feature participates</td> </tr> <tr> <td>ZLocation</td> <td>The value representing the depth being associated with the feature</td> </tr> <tr> <td>XLocation</td> <td>The x-coordinate of the feature for the given depth</td> </tr> <tr> <td>YLocation</td> <td>The y-coordinate of the feature for the given depth.</td> </tr> </tbody> </table>	MeasurementID	A unique identifier for the feature created by combining unique combinations of FeatureID and ZLocation	FeatureID	A key field relating to the FeatureID of the feature	FeatureClass	The name of the feature class in which the relating feature participates	ZLocation	The value representing the depth being associated with the feature	XLocation	The x-coordinate of the feature for the given depth	YLocation	The y-coordinate of the feature for the given depth.
	MeasurementID	A unique identifier for the feature created by combining unique combinations of FeatureID and ZLocation												
FeatureID	A key field relating to the FeatureID of the feature													
FeatureClass	The name of the feature class in which the relating feature participates													
ZLocation	The value representing the depth being associated with the feature													
XLocation	The x-coordinate of the feature for the given depth													
YLocation	The y-coordinate of the feature for the given depth.													

OBJECT CLASSES (cont'd)	TSType —An index of the types of time series data stored in the TimeSeries table.		
	Notes		
	Fields	TSTypeID	The identifier for a type of time series
		Variable	The type of data being described by the time series
		Units	The units of measurement
		IsRegular	Whether the units have a regular or irregular time step
		TSInterval	The interval between two time steps
		DataType	The type of time series. This is solved by the use of a coded value domain: 1 = Instantaneous 2 = Cumulative 3 = Incremental 4 = Average 5 = Maximum 6 = Minimum
		Origin	The origin of the time series. This is solved by the use of a coded value domain: 1 = Recorded 2 = Generated
	TimeSeries —Stores the time-varying attributes of features.		
	Notes		
	Fields	FeatureID	A key field for relating this time series to a feature class
		TSTypeID	A key field for relating this time series to an entry in the TSType table
		TSDatetime	The data and time stamp of a time series value
		TSValue	The measured value for the time step
	Cruise —Defines the characteristics of a ship for the duration of an expedition.		
	Notes		
	Fields	CruiseID	An identifier for a given cruise
		Code	A user-defined code for a given cruise
		Name	The name of the cruise
		Purpose	The purpose of the cruise
		Status	Defines the status of the cruise
		Description	A general description of the cruise
StartDate		The beginning time stamp for the cruise	
EndDate		The ending time stamp for the cruise	
ShipName		The name of the ship participating in the cruise	

OBJECT CLASSES (cont'd)	MeasuredData —Stores recorded values for a given parameter.		
	Notes		
	Fields	MeasurementID	A key field for relating this table to the Measurement table
		ParameterID	A key field for relating this table to the Parameter table
		DeviceID	A key field for relating this table to the MeasuringDevice Table
		DataValue	The recorded value
	MeasuringDevice —Stores information pertaining to the device taking the measurements.		
	Notes		
	Fields	DeviceID	A key field for relating this table to either another table or feature class
		Name	The name of the measuring device
		Description	A description of the measuring device
		VehicleID	A key field relating this table to the Vehicles table
	Parameter —Stores information about the parameters being measured.		
	Notes	This table can be used as a mechanism for querying a geodatabase for a specific parameter and then finding values of a particular type in related tables. Or it can be used as a lookup table of parameter types for a particular value.	
	Fields	ParameterID	The unique identifier of a specific parameter
Name		The name of a parameter	
Description		The description of a parameter	
Quantity		The quantity type for a parameter. This is solved by the use of a coded value domain: 1 = Other 2 = Scalar 3 = Vector	
Unit		The unit of measure for a parameter	
Significant Digits		The number of significant digits defining the precision of this parameter	

RELATIONSHIPS	CruiseHasTracks	1 : *	One Cruise can have zero or many Tracks
	SurveyInfoHasPoints	1 : *	One Survey can have zero or many points
	MeasurementHasData	1 : *	One Measurement can have zero or many MeasuredData values
	MeasuringDeviceHasData	1 : *	One MeasuringDevice can have zero or many MeasuredData values
	ParameterHasData	1 : *	One Parameter can have zero or many MeasuredData values
	TSTypeHasTimeSeries	1 : *	One TSType can have one or more TimeSeries

References

- Arctur, D., and M. Zeiler. 2004. *Designing geodatabases: Case studies in GIS data modeling*. Redlands, Calif.: ESRI Press.
- Hennessy, M., D. Smyth, and T. Alcorn. 2006. *Building a spatial data warehouse for the Marine Institute*. Unpublished Technical Report. Galway, Ireland: Marine Institute.
- Langran, G. 1992. *Time in geographic information systems*. London: Taylor & Francis.
- Maidment, D., V. Merwade, T. Whiteaker, M. Blongewicz, and D. Arctur. 2002. Time series. In *Arc Hydro: GIS for water resources*, ed. D. Maidment, 141–66. Redlands, Calif.: ESRI Press.
- Marine Institute. 2005. Ireland's ocean economy and resources. *Marine Foresight Series 4 Technical Report*. Galway, Ireland: Marine Institute.
- Miller, H. J. 2005. A measurement theory for time geography. *Geographical Analysis* 37(1): 17–45.

Case study content

Dawn Wright, Oregon State University

Michael Blongewicz, DHI Water & Environment

Martina Hennessy, Damian Smyth, and Trevor Alcorn, Marine Institute, Galway, Ireland

Rob Morrison, ESRI Ireland