Prometheusplein 1
Postbus 98
2600 MG DELFT
Telefoon: 015 - 2784636
Fax: 015 - 2785673
Email: Klanten-Service@Library.TUDelft.NL

Datum: 07-apr-03

Bonnummer: 648366

Aan: VLAAMS INSTITUUT VOOR DE ZEE
VISMIJN
PAKHUIZEN 45-52
B 8400 OOSTENDE
BELGIE

Tav:

Aantal kopieën: 20

Uw referentie(s): 1521553 1521553

Artikelomschrijving bij aanvraagnummer: 648366

Artikel: Developing a Marine Information System by Integrating Exis
Auteur: A Jacob et al.
Tijdschrift: MARINE GEODESY
Jaar: 2003 **Vol.** 26 **Aflevering:** 1/2
Pagina(s): 87-106 **Plaatsnr.:** GEPR

Met ingang van 1 april 2003 zullen de prijzen voor fotokopie levering buitenland stijgen met € 0,05 per pagina

From April 1 2003, prices for photocopy delivery abroad will increase by € 0.05 per page

Marine Geodesy, 26:87–106, 2003

0149-0419/03 $12.00 + .00
DOI: 10.1080/01490410390181199

Taylor & Francis
Taylor & Francis Group

Developing a Marine Information System by Integrating Existing Ocean Models Using Object-Oriented Technology

A. JACOB

Merck, Sharp and Dohme
Granville, Australia

T. HAMRE
G. EVENSEN

Nansen Environmental and Remote Sensing Center (NERSC)
Bergen, Norway

K. A. MUGHAL

Department of Informatics
University of Bergen
Bergen, Norway

Information systems developed for different applications within the environmental domain have common characteristics, which can potentially be abstracted for sharing and reuse of design and software modules. This article presents an approach to designing for reuse by abstracting commonalities in the design of a Marine Information System (MIS), facilitating data management in a prototype operational monitoring, forecasting, and management system for the North Atlantic and the Nordic Seas. A detailed study of the requirements and data analysis was carried out, and Object-Oriented Technology (OOT) is employed to encapsulate abstractions and to promote reuse of code and design. This article identifies the Object-Oriented Frameworks (OOFW) required to build the MIS. It also provides guidelines to environmental scientists for restructuring legacy software and employing modern programming techniques.

Keywords marine information system, object-oriented technology, design and software reuse, data assimilation, ocean model, ecosystem model

The need for better monitoring and prediction systems for marine phenomena and processes has been demonstrated by the failure of government and research establishments to anticipate and prepare for a number of catastrophic events such as storm surges, harmful algal blooms, and oil spills which have occurred during the past few years. The implementation

Received 27 February 2002; accepted 20 June 2002.

The work presented in this article was funded by the Norwegian Research Council (NFR) under contract 119190/410 and a project entitled "Strategy for promoting recruitment within the IT field" funded by NFR and the University of Bergen. The modeling and assimilation is possible due to the support received from the Norwegian Supercomputing Committee through a grant of computing time. We also gratefully acknowledge Dr. P. Samuel and M.Sc. M. Bentsen for their comments and suggestions.

Address correspondence to Torill Hamre, Nansen Environmental and Remote Sensing Center, Edvard Griegsvei 3a, N-5059 Bergen, Norway. E-mail: torill.hamre@nersc.no

of an operational monitoring and forecasting system will enable early warning of environmental hazards. It will also facilitate timely initiation of measures to contain or reduce the impact of such hazards, as well as improve our understanding of the marine ecosystem for efficient resource management.

The DIADEM (not an acronym) project funded by the European Commission (Evensen 1998) proposes to develop an MIS prototype for such an operational monitoring, forecasting, and management system for the North Atlantic and Nordic Seas. The proposed system will combine physical, biological, and chemical data from in-situ measurements and remote sensing with a coupled physical and ecosystem model through the use of data assimilation techniques. The system will provide optimal simulations of the spatial and temporal distributions of the key variables characterizing the ocean system.

The existence of a large number of sensors for monitoring the environment, with inherent differences in spatial and temporal resolutions and formats, makes it necessary to have a sophisticated computer tool to manage this large volume of heterogeneous data and prepare them for assimilation into models. The MIS has been proposed to meet the data administrative and integration requirements of this system. In addition, the MIS will provide a user friendly environment that facilitates setting up and running the models, allowing rapid evaluation of different alternatives as well as facilitating rapid set up of different hybrid models. By defining the workflow, the MIS will lead the user through the various steps for setting up and running the available prediction models.

In this article, some aspects leading to the design and development of the MIS tailored to meet the requirements of the DIADEM project are discussed. The following section analyses the various features of the DIADEM system, including the models, data assimilation, and the data sets. A later section introduces the reuse techniques and methods used for designing the MIS. Various design details are presented next. These include the overall architecture and the Frameworks identified. Next, a section looks at other, related OO Information Systems (IS) and lessons learned there. The article concludes with a look at what has been done so far and the future work to be undertaken.

The DIADEM System Features

The DIADEM project will provide a prototype of an operational system for the North Atlantic and the Nordic Seas capable of monitoring and predicting the three-dimensional distribution and evolution in time of a set of physical and ecosystem variables, such as temperature, salinity, density, current velocity, chlorophyll, nutrients, and inorganic carbon. At the core of the proposed system is a three-dimensional, primitive equation, physical ocean model coupled with a marine ecosystem model. The model is initialized with NODC (National Oceanographic Data Center) data or modeled temperature and salinity data, and forced by realistic meteorological data. In order to make the model simulations more reliable, the coupled model will assimilate observations derived from satellite altimeter and radiometers. The main DIADEM system components include the models, the assimilation techniques, and the data sets (see Table 1). Each of these modules will be expanded upon in the following sections. The data flow shown in Figure 1 illustrates how the different data types, namely, remote sensing, model, and in situ (data), undergo quality control and various processing steps before being assimilated into the model.

The Coupled Ocean-Ecosystem Model

Numerical models are used to simulate various aspects of the ocean environment by numerically solving the various equations governing the behavior of the system. Physical

TABLE 1 The Main Modules of the DIADEM System

MIS parts	Modules	Contents
1. Model	• MICOM physical model • FDM marine ecosystem model	
2. Assimilation	• Ensemble Kalman filter • Ensemble smoother • Singular evolutive extended Kalman filter • Multivariate optimal interpolation methods	
3. Data sets Initial fields	NODC	➢ Temperature ➢ Salinity
	Physical model	➢ Temperature ➢ Salinity ➢ Cloud cover ➢ Relative humidity ➢ Precipitation
	Ecosystem model	➢ Nutrients ➢ Biomass ➢ Phytoplankton ➢ Zooplankton ➢ Bacteria
Forcing fields	Climatological and synoptic	➢ Wind vector ➢ Wind speed ➢ Air temperature ➢ Cloud cover ➢ Relative humidity ➢ Precipitation
	Remote sensing observations	➢ Sea surface height (altimeter) ➢ Sea surface temperature (infrared) ➢ Chlorophyll-a (radiometer)
Output data	Physical variables	➢ Temperature ➢ Salinity ➢ Sea surface height ➢ Current velocities
	Biological-chemical variables	➢ Chlorophyll-a ➢ Inorganic carbon ➢ Nutrients ➢ Dissolved organic substances

ocean models simulate the response of the system to physical forcing terms, such as wind, solar radiation, river inflow, and so forth, while biological models simulate the response of the ocean biota to changes in the physical, chemical, and geological variables, as well as their mutual interactions. Combined physical and biological models can be used to understand, monitor, and predict the spatial and temporal changes of marine ecosystems.

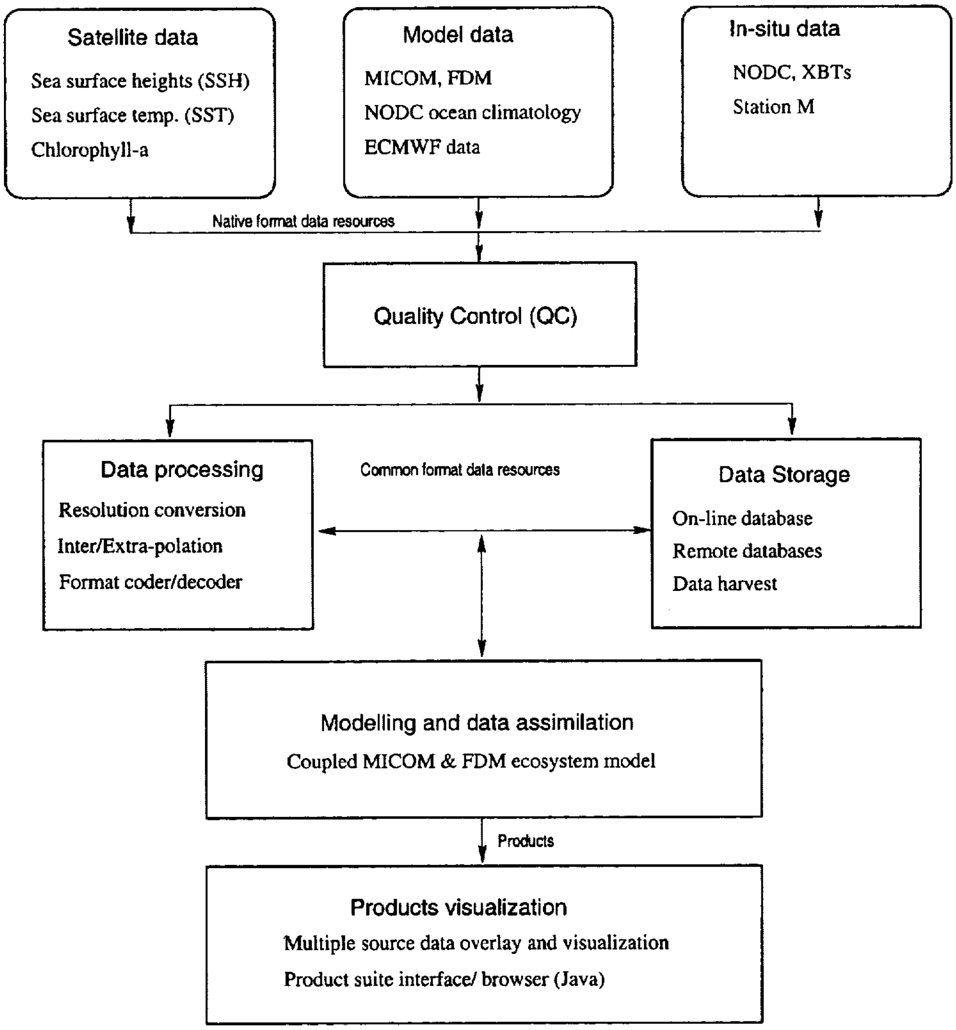

FIGURE 1 Data flow in DIADEM.

The Miami Isopycnic Coordinate Ocean Model (MICOM)

The physical model used is an adaptation of the MICOM, which is an isopycnic dynamic-thermodynamic Ocean General Circulation Model (OGCM) developed by Bleck and colleagues (1992). The model is driven by atmospheric fluxes, bottom topography, wind, and fresh water flux. It allows for realistic descriptions of the land and bottom boundaries and provides simulations of the three-dimensional distribution and temporal evolutions of parameters such as temperature, salinity, current velocity, and density. The MICOM model has been further developed at NERSC to include new thermodynamic routines for computation of surface fluxes, and it has also been coupled to a dynamic-thermodynamic ice model. A simplified schematic is shown in Figure 2. The MICOM model is implemented in the Fortran90 programming language.

The Fasham, Ducklow, and McKelvie Ecosystem Model (FDM)

A slightly modified version of the 3-D ecosystem model described by Sarmiento and others (1993) has been coupled to the MICOM model. The model shown in Figure 3 is

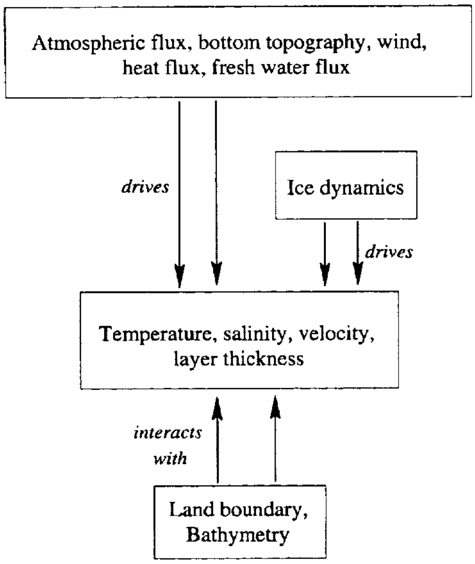

FIGURE 2 The MICOM physical model.

a seven-compartment model with phytoplankton, zooplankton, and bacteria as the living biota, and the nutrients ammonium and nitrate. In addition, there is one pool of dissolved organic nitrogen (DON) and one pool of particulate organic nitrogen (detritus).

The model is based on nitrogen as the currency of the ecosystem model. It is formulated in two modules, one that describes the exchange of nitrogen between compartments in the euphotic zone, and the other that parameterizes the decay of particulate and dissolved organic nitrogen below the euphotic zone. The ecosystem is only explicitly modeled in the euphotic zone; below the euphotic zone, all the organic matter is gradually turned over to ammonium and then to nitrate. The FDM model is implemented in Fortran77 and Fortran90.

Common Model Set-Up Features

The orthogonal curvilinear grid. A conformal mapping is used to define new grids (Bentsen et al. 1999), with the locations of the North and South poles mapped to two arbitrary locations on the Earth (see Figure 4 for an example). This remedies the problem of the grid singularity at the North pole, which would be the case if using a traditional latitude-longitude coordinate system.

The coordinate transformation consists of several steps. First, the spherical coordinates are mapped to the extended complex plane by stereographic projection. The next step is a conformal mapping in the complex plane, which is accomplished with a linear fractional transform. This particular mapping has the desired properties: that circles are mapped to circles, new locations of the poles are easily determined, and an inverse exists. The final step utilizes the inverse stereographic projection to get from the complex plane back to spherical coordinates. The transformation between spherical coordinates has the following important properties: orthogonality is preserved; circles are mapped to circles; the shape of infinitesimal figures are preserved; and analytical expressions exist to move back and forth between the spherical coordinate systems.

The conformal mapping can be prepended by a Mercator transformation, yielding a grid that locally has the same metric scale factors in both directions. By using the additional Mercator step, stretched grid cells towards the poles can be avoided. Whether or not a Mercator transformation is added to the conformal mapping, the location of the poles can

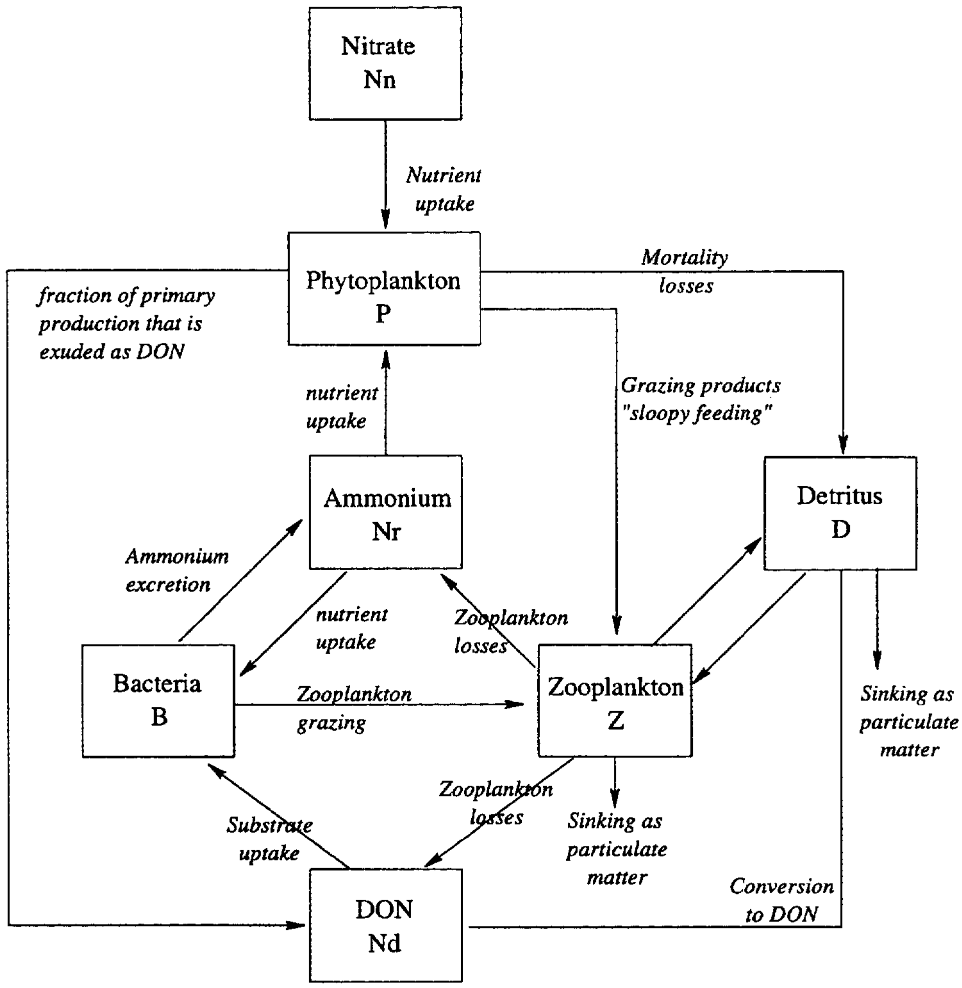

FIGURE 3 The ecosphere model for the euphotic zone (Drange 1994).

be chosen so that the resulting grid has an enhanced resolution between the poles. This allows for the use of a large model domain where high resolution is still retained in the area of interest.

The increase of resolution between the poles is dependent on the distance between the poles. For the first test experiments in DIADEM, with poles on the North American and European continents, the resolution between the poles was about 50 km. Currently, the grid resolution is about 30 km in the Gulf Stream, about 60 km in the Arctic Ocean, and about 90 km in the Southern Atlantic Ocean. If poles are placed more closely together, for example, in central Europe and on the NW African continent, the grid resolution would vary from about 10 km along the West African coast to about 1° along the South American coast (Bentsen et al. 1999).

The use of a variable model resolution and a larger model domain also allows "closed" boundary conditions far from the area of interest where a relaxation to climatology is applied. The grid transformation routines are implemented in Fortran77 and Fortran90.

Bathymetry. The bathymetry is interpolated to the model grid using an improved version of the ETOPO-5 global bathymetry data set named DS759.2 from the Terrainbase

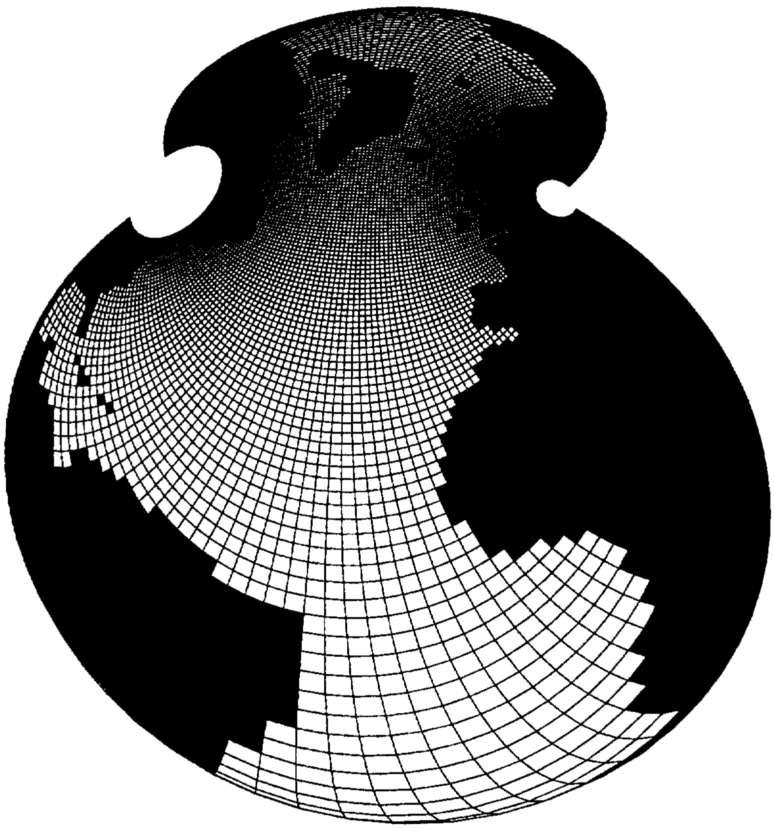

FIGURE 4 Example of an orthogonal curvilinear model grid. Note the very high resolution in the North Atlantic and the Nordic Seas compared to the Arctic and the Southern part of the Atlantic.

Project conducted by the National Geographic Data Center and World Data Center (Anon. 1998a). Averaging is used if there are more than four data points within a grid cell. Otherwise, bilinear interpolation between the two closest data points is used to estimate the depth in the current grid cell. The averaging method is equivalent with a first-order conservative remapping scheme, as described in Jones (1999), which also addresses second-order conservative remapping schemes. The interpolation routines are implemented in C, Fortran77, and Fortran90.

Assimilation

A numerical ocean model is a computer program that uses a description of the current situation of a phenomena or process, together with some knowledge of its behavior, to estimate how it is likely to evolve over time. The result is a time series of values for a set of parameters describing the state of the ocean at a number of time steps between the initial conditions and a chosen end time. However, due to unavoidable errors and approximations present in the mathematical and numerical formulations, every numerical model will drift away from the observed state, and after a while, the results will be worthless for prediction or monitoring purposes. Data assimilation attempts to rectify this problem in numerical modeling by using a limited set of direct observations to check the drift of the model simulations from actual observations.

Data assimilation combines information from observations (remote sensing and in-situ) and numerical forecast models in an optimal manner. A typical data assimilation system is composed of a forecast model that provides a background or "guess" state of the ocean by extrapolating information forward from previous observations, and an analysis that combines information from the background and from the entire set of widely scattered observations into a model-compatible form for extrapolation to the next analysis time. Insertion of observations in the analysis may be either periodic (e.g., every six hours) or continuous. The analysis procedure should be capable of combining different measurements (e.g., of wind, sea surface temperature, etc.) from a variety of instruments with differing spatial and temporal resolutions and inherent error characteristics.

The various data assimilation schemes listed in Table 1 aim at predicting error statistics where an ensemble of ocean states is integrated forward in time, and the error statistics needed to perform a variance minimizing analysis can be calculated from the ensemble (Evensen 1992, 1994). The assimilation schemes are implemented in Fortran77 and Fortran90.

Data Sets

Initial and Boundary Value Fields

A numerical model run requires that the variables simulated by the model be specified at all grid points for the initial time step from which the simulation is to take place (initial conditions) and at all the grid points along the boundary of the area for which the simulation is carried out (boundary conditions).

In the proposed system, initial fields are obtained from the NODC or the global Levitus data sets (Table 2), consisting of globally gridded climatological values of temperature

TABLE 2 Description of Global Data Used to Initialise the MICOM Model (A/S/M Stands for Annual, Seasonal, Monthly)

Source	Variable	Periodicity A/S/M	Resolution	Remarks
Levitus	Sea surface temperature	M	1° × 1°	at 32 depths
	Ocean temperature	A	1° × 1°	
	Salinity	A	1° × 1°	
	Sea surface salinity	A, M	1° × 1°	
NODC	Ocean temperature	A, M, S	1° × 1°	at 19 depths
	Salinity	A, S, M	1° × 1°	
ECMWF	Air temperature	M	2.5° × 2.5°	1985–1989 average
	Wind speed	M	2.5° × 2.5°	1985–1989 average
	Std. Var. wind speed	M	2.5° × 2.5°	1985–1989 average
	Wind components	M	2.5° × 2.5°	1985–1989 average
COAD	Cloud cover	M	2° × 2°	no polar data
	Relative humidity	M	2° × 2°	no polar data
Legates	Precipitation	M	1° × 1°	
ECMWF	Air temperature	Synoptic	1° × 1°	1979–1996
	Wind speed	Synoptic	1° × 1°	1979–1996
	Wind components	Synoptic	1° × 1°	1979–1996
	Sea level pressure	Synoptic	1° × 1°	1979–1996
	Relative humidity	Synoptic	1° × 1°	1979–1996
	Cloud cover	Synoptic	2.5° × 2.5°	1979–1996

and salinity at a number of depths. On open boundaries, a relaxation of temperature and salinity of water mass properties to monthly climatological values is normally used. Alternatively, model results from a global OGCM, that is, a global spin up data set, may be used to initialize the regional model. At the open boundaries, a relaxation of temperature and salinity to monthly averaged climatologies produced by the global model is applied.

Forcing Fields

The MICOM model interacts with the atmospheric forcing through a mixed layer. The water in the mixed layer varies due to transfer of momentum from the atmospheric winds and by thermodynamic forcing through fresh water, heat, and buoyancy fluxes. The momentum fluxes are computed from wind stress and the fresh water fluxes from precipitation, evaporation, surface and air temperature, and cloud cover. The surface forcing in DIADEM is done using *climatological* (seasonal or monthly averages) data such as ECMWF (European Center for Medium Range Weather Forecast) wind field (vectors), ECMWF wind speed, ECMWF air temperature, COAD (Comprehensive Ocean Atmospheric Data) cloud cover, COAD relative humidity, and Legates & Wilmott precipitation and *synoptic* (6 hourly or 12 hourly, etc.) data (Legates and Wilmott 1990) such as the ECMWF reanalysis data (air temperature, sea level pressure, and relative humidity) available for the period 1979–1993 (refer to Table 2).

Remote sensing observations. Altimeter Sea Surface Height (SSH) data determined from the TOPEX-POSEIDON and the European Remote Sensing (ERS) satellite series, the Advanced Very High Resolution Radiometer (AVHRR) Sea Surface Temperature (SST) data from the National Oceanic and Atmospheric Administration (NOAA) Pathfinder series, and chlorophyll pigment concentration derived from the Coastal Zone Color Scanner (CZCS) and Sea-viewing Wide Field-of-view Sensor (SeaWiFS) are analyzed, processed, and prepared for direct assimilation into the model (see Table 3).

Estimates of the dynamic SSH can be obtained from satellite radar altimeters that observe the sea surface topography along a ground track covering the entire global ocean at regular intervals. The data are typically provided at 7 km intervals along the ground track, but include large data gaps between adjacent passes of the ground track. The data processing includes selecting and editing the data, application of a large number of instrumental and environmental corrections usually obtained from external models, least squares techniques to remove orbital error, subtraction of the temporal mean to correct for the geoid height, and interpolation to an appropriate regular spatiotemporal grid. In DIADEM, altimeter data from ERS-1/2 and TOPEX/POSEIDON satellites from 1992 onwards are used for assimilation as well as validation.

Estimates of the SST are derived from the brightness temperatures measured by AVHRR instruments on board the NOAA-12, NOAA-14, and NOAA-15 satellites. The AVHRR thermal infrared channels measure the radiation in the 3 μ to 12 μ waveband and have a

TABLE 3 Data and Sensors Used in DIADEM Assimilation

Data	Instrument	Satellite	Availability
Sea Surface Height (SSH)	Altimeter	TOPEX-POSEIDEN	1992–1996
		ERS	1992–1996
Sea Surface Temperature (SST)	AVHRR	NOAA Pathfinder	1992–1996
Ocean color	CZCS	Nimbus-7	1978–1986
	SeaWiFs	OrbView-2	1997–present

TABLE 4 Overview of Available Ocean Colour Data (Periodicity A, S, M, D Stand for Annual, Seasonal, Monthly, Daily; LAC and GAC Stand for Local Area Coverage and Global Area Coverage, Respectively)

Ocean colour sensor	Variable	Resolution	Coverage	Periodicity
CZCS	Concentration of chlorophyll-a + phaopigments	0.85 km LAC; 4.5 km GAC	Global; No polar data	A, S, M, D
SeaWiFS	Concentration of chlorophyll-a	1.13 km LAC; 4.5 km GAC	Global; No polar data	A, S, M, D

resolution of 1 km, 0.2°K. The radiation measured by this instrument has to be corrected for absorption by water vapor and gases in the atmosphere (McClain et al. 1983).

The main product derived from ocean color remote sensing sensors is the integrated chlorophyll-a (Chl-a) concentration of the water column, limited to the upper ocean layer (see Table 4). In order to assimilate satellite data into the numerical model, the phytoplankton biomass has to be calculated from the chlorophyll concentration, and the subsurface chlorophyll concentration has to be derived from the satellite-derived surface value.

Example of DIADEM Model Output

During the initial phases of DIADEM, test experiments were run to demonstrate and validate the models and assimilation schemes. Figure 5 shows an example of observed and forecasted SST for day 184 of 1996. The satellite observations of SST are from the AVHRR sensor and have been quality controlled and gridded before being assimilated in the ocean model. AVHRR data for a period of 7 days were used to generate a gridded SST data set, using optimal interpolation where the most recent data are given the highest weight. The DIADEM MICOM model was run with a model set-up as described earlier and generated the SST forecast shown in the lower part of Figure 5. Model predictions are generally in good agreement with the observed SST data, but some smaller scale features are not fully resolved due to the rather coarse grid resolution.

Since the first demonstration cases, the DIADEM models have been continuously developed, leading to both significant increases in grid resolution and extended data assimilation schemes. A new EU-funded project, TOPAZ, continues this development of an operational ocean monitoring and prediction system for the North Atlantic, the Nordic Seas, and the Arctic Ocean, using state of the art numerical model tools and data assimilation methodologies. In DIADEM, the MICOM model (Bleck et al. 1992) was used together with a three-dimensional implementation of the ecosystem model by Fasham et al. (1990). Remotely sensed sea-surface heights and temperatures were assimilated in the physical model, while ocean color data were used in a demonstration experiment for assimilation in the ecosystem model. Recently, the MICOM has been replaced with the Hybrid Coordinate Ocean Model (HYCOM) described by Bleck (2002). In addition the system is now assimilating satellite measurements of sea ice concentration and in situ profiles of temperature and salinity.

Future development of the model systems will focus on the establishment of a forecasting capability for nested regional models, which can be used to support offshore industry operating along and off the continental shelf. Thus, the long-term objective is to provide realtime ocean products particularly customized for the offshore oil industry. Object

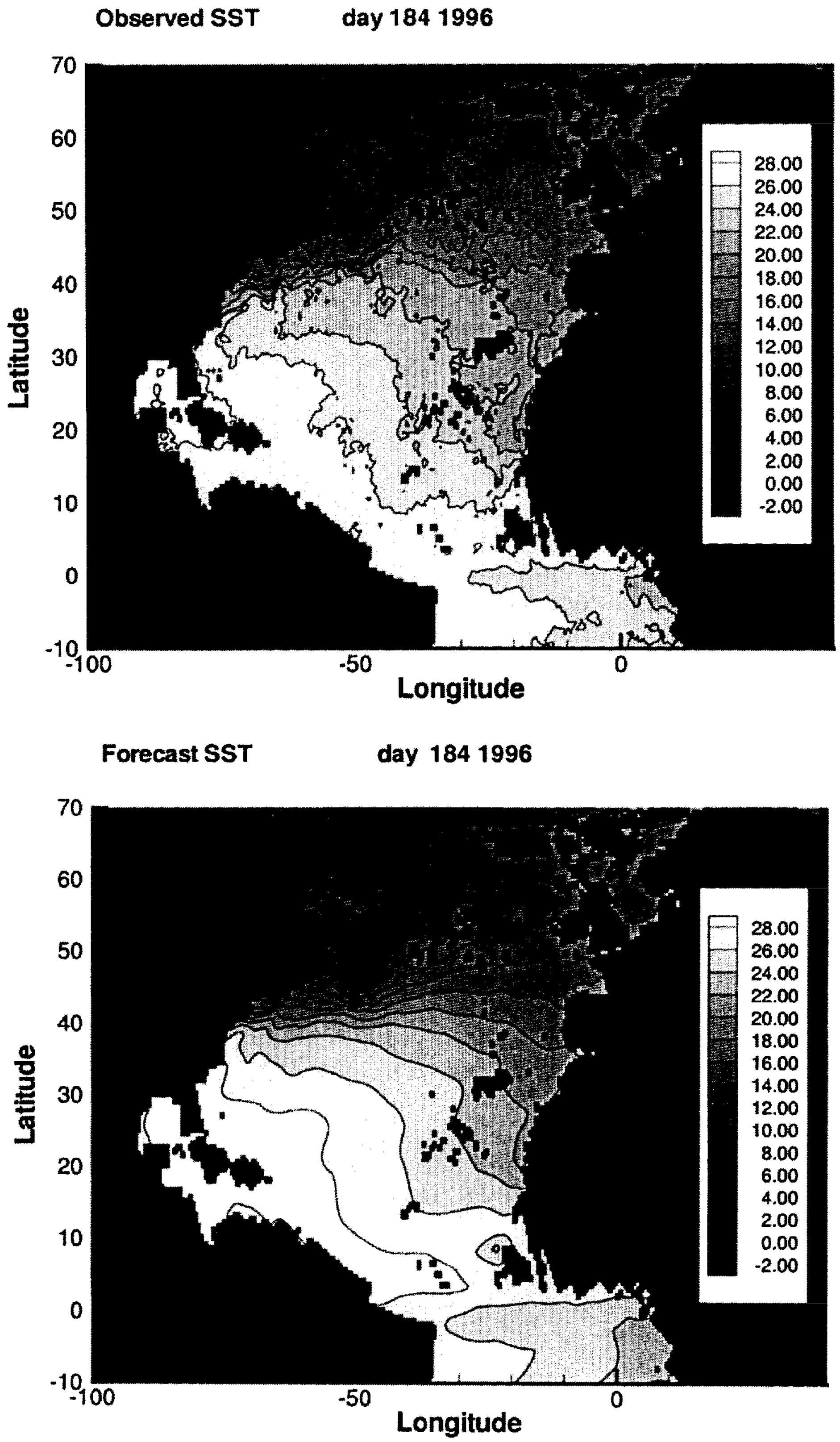

FIGURE 5 Observed SST (top) and forecasted SST (bottom) for day 194, 1996.

technology can play an important role in this development by offering mechanisms for seamless system integration and low cost delivery of products to users via Internet. Reuse is the driving force behind the MIS development for ocean monitoring and prediction. In particular, reuse of existing software will be essential to retain a stable production environment.

Designing for Reuse

The potential for reuse is present at all stages of software development: analysis, design, implementation, and testing. Functions, abstract data types, and libraries were the earliest form of code reuse. With a shift in paradigm to OOT, it was found that design reuse is more important and more difficult than code reuse. Design patterns and the Unified Modelling Language (UML) all promote design reuse, whereas objects and Frameworks promote reuse of design and code. The MIS is designed with reuse in mind from the very beginning—reuse within the system, reuse within the domain, and reuse across domains.

Design Reuse

UML (Rumbaugh et al. 1999) provides a set of graphical notation conventions that a developer can use to describe or model an application. UML provides five different groups of UML diagrams—Use Case diagrams, Static Structure diagrams, Interaction diagrams, State diagrams, and Implementation diagrams—that allow the user to analyze a problem from different perspectives (Larman 2001). UML consists of a meta model in addition to the graphical notation. The UML meta model contains elements for customizing it to specific application areas, as well as for extending the UML notation. This makes UML far more powerful than other static Object Oriented Analysis and Design (OOAD) methodologies, where the ability to model a system is determined by the capabilities of a fixed notation. The application specifc models can be reused within the domain. The MIS is conceptually modeled using UML.

Design patterns identify, name, and abstract recurring problems in OO design and provide a practical solution for these. They capture the intent behind a design by identifying objects and analyzing how objects interact and how responsibilities are distributed among them. They constitute a base of experience for building reusable software. Thus, design patterns are higher levels of abstraction than classes or objects, and they act as building blocks from which more complex designs can be built (Buschmann et al. 1996; Gamma et al. 1995). Patterns are used in the design of the MIS whenever a problem arises that has a proven solution and in this way adds to the reliability of the software system. Of particular interest is discovering new patterns that are specific to the MIS domain.

Design and Code Reuse

In object-oriented software development, systems are organized as a collection of cooperating entities, called *objects*, which include both data structure and behavior. Objects representing the same phenomena or features in the real world are grouped in *classes*, which define the common data (attributes) and behavior (methods) (Rumbaugh et al. 1991). Objects demonstrate the properties of abstraction, encapsulation, inheritance, aggregation, polymorphism, and dynamic binding. Each of these properties individually and synergetically promotes modular and reusable programming. Thus objects embody both the blueprint and implementation. We would like to leverage the advantages that OO paradigm offers in the design and development of the MIS.

A Framework (FW) is a collection of abstract and concrete classes that provides the skeleton for a system or part of a system in a given domain, and that can be customized to fit the needs of a particular client in the same domain or in another domain. They provide the code fragments as well as the interfaces that act as the "glue" to connect to other software components (Adair 1995). Users of the FW can extend or customize the FW to suit their particular application, thereby reusing both code and design.

Frameworks are extended or customized using inheritance and/or composition. Frameworks are also classified as horizontal FWs (generic) and vertical FWs (specific) (Demeyer et al. 1997; Rogers 1997; Schmid 1997). The MIS is defined in terms of FWs that abstract common functionality, thereby promoting a modular and flexible style of programming and also reusability of the FW in other similar Information Systems (IS).

Code Reuse

Components are executable pieces of reusable software that provide their services through well defined interfaces. Further, a component can be deployed independently and is subject to composition by third parties (Szyperski 1998). Building software from components means creating applications in whole or in part from existing pieces. Thus components are designed to facilitate reuse of code. Components do not have to be object oriented; they can be realized using the functional programming approach, using assembly language, or using any other approach. OOFWs evolve into components if customized with concrete implementations. Parts of the MIS are very specific to the ocean prediction and monitoring system and can be implemented as software components.

A legacy system is an existing software system, often developed using a non-OO programming language. For historical reasons, such systems frequently rely on proprietary data formats and/or nonstandard communication protocols, with lacking or inadequate documentation. This makes legacy systems difficult to integrate with other systems. Nevertheless, existing software systems are used in new systems because

- They are successful pieces of software with substantial amount of code that have been tested, debugged, and used over the years.
- Porting legacy systems to the new (OO) system is time-consuming, as the resulting code will have to be tested and debugged again.
- Number crunching and time-consuming algorithms, which are the norm in numerical modeling and image processing, typically have fast implementations in C or Fortran90.
- The legacy software might be dependent on libraries (e.g., mathematical libraries) or system features (e.g., multiprocessors), to such an extent that conversion to the current or planned system will be cumbersome and even impossible.

A native method provides the means to call a method that is implemented in a language different from the calling method or implementation. The reasons for using native methods are the same as for using legacy systems, the most compelling being accessing the underlying operating systems, accessing existing libraries, and optimizing performance. Most languages provide some mechanism to call routines written in another language, such as the Java™ Native Interface (JNI) in Java™ (Liang 1999). Various algorithms and data processing routines for gridding and image processing, for example, are implemented in C and Fortran programming languages. These routines are examples of methods that are accessed by the native methods of the object-based MIS.

Design Details

In designing the MIS for the DIADEM project, only the data, tools, and models required for ocean monitoring and prediction for the North Atlantic and Nordic Seas will be "active" within the MIS. This will include the coupled ocean-ecosystem model, oceanographic and meteorological data, in situ data from weather stations, and remote sensing data as listed in Table 1.

A software system that integrates models, data(bases), and decision-making tools within the environmental domain, and packages them in a way that makes them useful for different types of users is called an Environmental Decision Support System (EDSS) in Rizzoli and Young (1997). The MIS satisfies this definition and is indeed an EDSS. The system proposed here is a *single level system* in that it is developed for the environmental scientist only. At a later stage, this single level system can be extended to a two-level system by plugging in a different user interface for decision makers.

The MIS proposed here can also be categorized as a *problem-specific EDSS* as opposed to a situation- and problem-specific EDSS. The MIS is used to monitor and forecast physical, biological, and ecosystem parameters for theoretically any oceanic region of the world, provided sufficient observational data are available for model parameterization. The MIS is not developed keeping a specific geographic location in mind; instead, it can be easily customized for new locations on the globe.

MIS Definition

The data management, analysis, manipulation, and presentation functions, as well as user interface, are taken care of by the MIS. The MIS proposed is an information system offering facilities for both data management and routine processing, as well as powerful tools for analysis, simulation, and integration (Hamre 1995). Large volumes of heterogeneous data are available from different sensors at different resolutions in space and time, which users need to use either singly or in combination (synergetically). The MIS is a system for providing a seamless integration of diverse data and associated tools to enable users to locate and process relevant data sets in a timely manner. The MIS includes: (1) tools for data storage and maintenance; (2) capabilities for selection of data based on their spatial, temporal, and/or thematic properties; (3) tools for integration of diverse data (remote sensing data, field observations, and model results); (4) tools for data analysis and visualization; and (5) simulation models for prediction of marine parameters (see Figure 6).

Overall Framework Architecture

Any IS has essentially three interacting modules: input, processing, and output. The MIS is a typical IS and is configured using the Model-View-Controller (MVC) architecture (Gamma et al. 1995). The model components encapsulate the data and processing; the view components encapsulate the display of information or the output; and the controllers translates user requests for the model or for the view component. The view and controller components together constitute the user interface. Figure 7 shows the top-level architecture of the MIS.

Framework Description

The top level architecture is further refined into smaller FWs, which are easier to understand, develop, and maintain, as shown in Figure 8. The bottom row of FWs corresponds to the

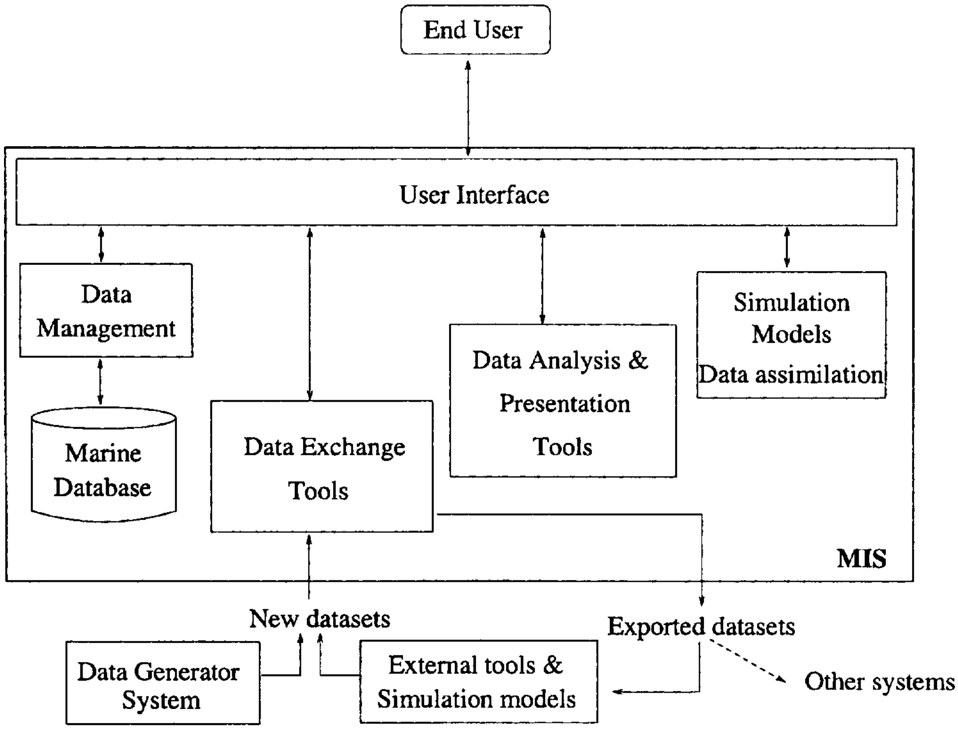

FIGURE 6 Components of the Marine Information System.

Model component in the MVC architecture, whereas the top two together correspond to the user interface, which is made up of View and Controller components. The controller component provides the context manager and the requisite application code required to hold the FWs together.

The FWs identified here encapsulate the common and repeated functionality, leaving stubs for extension and customization. These FWs have been identified during the initial iterations. As these FWs are used in developing similar applications in the environment domain, the FWs will evolve to give rise to more generic and robust FWs.

View and controller components FWs. The Graphical User Interface (GUI) FW is a generic horizontal FW that provides all the functionality for interacting with the client of the MIS and for visualizing the results. Several GUI tools are available, such as the Swing components from Java (Geary 1999), ET++, and Motif. GUI was one of the earliest

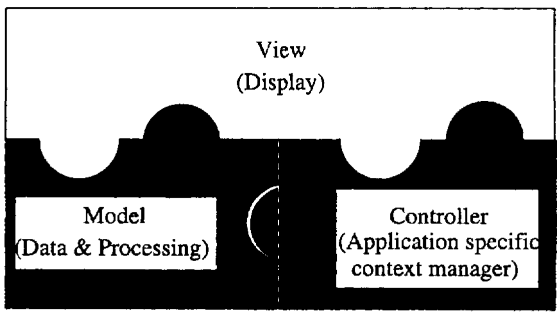

FIGURE 7 MIS top level FW architecture.

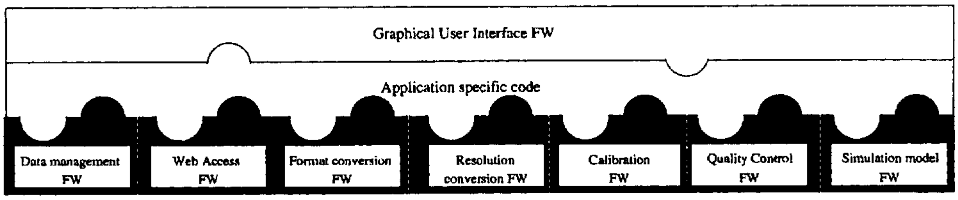

FIGURE 8 The Frameworks used for building the DIADEM MIS.

Frameworks to be developed and the one most extensively dealt with. The GUI for the MIS is developed using the Java Swing API.

Model components FWs. The Data Management FW is a generic horizontal FW that abstracts all the data storage and data management functionality. Functionality such as storing, querying, retrieving, deleting, and modifying 1D, 2D, and 3D data, and in this case, spatiotemporal data, that is, data associated with a geographic location and time. This functionality is essentially provided by relational database management systems (e.g., Oracle, Sybase), object-oriented database management systems (e.g., ObjectStore, O2), or object-relational database management systems (e.g., Informix, Oracle 8). As this functionality is already available off the shelf, it will not be addressed in this article.

The Web Access FW is a vertical FW that is customized to access remote data sites and remote processing tools and models. This FW is introduced in the design to allow for a future distributed application and to provide for the inevitable shift from the Graphical User Interface (GUI) to the Network User Interface (NUI), as a result of the Internet.

The Format Conversion FW is a generic horizontal FW that converts data in one format to another format. The design and implementation of this FW has been addressed in some detail in Jacob (1999). Algorithms to convert between standard formats exist in different functional programming languages. For example, conversion software for standard formats such as NetCDF (Rew et al. 1996) and HDF (Anon 1998b) can be plugged into the FW. Java has released the Java Advanced Imaging (JAI) Application Programming Interface (API) (jai 2002) that enables incorporation of image processing facilities, including I/O for a set of standard formats, in Java applications and applets. In addition to handling these standard formats, the FW allows conversion routines for ad hoc and in-house data formats to be included.

The Resolution Conversion FW is a generic horizontal FW that converts data with a given resolution to another resolution. This is quite common in image processing and gridded data, but also required for vector data. Resolution conversion requires access to interpolation and extrapolation routines that are typically implemented in C and Fortran programming languages.

The Calibration FW is a vertical FW that is used to calibrate data from instruments, sensors, and models. Calibration is very specific to the application and the data sets used. Calibration is particularly important when similar parameters derived from different sensors have to be used together.

The Quality Control FW is a generic horizontal FW that is used to check the accuracy, consistency, and completeness of data. While standard quality controls can be built into the FW, such as consistency checks for data fields, ad hoc quality controls such as eliminating redundant data, ensuring that data values fall within predefined domain values, and specific quality criteria defined by acceptable values have to be tailored to suit the application.

The Simulation model FW is a vertical FW that sets up and initiates a model run for a particular application. The model FW provides a workflow procedure that guides a user through the steps for setting up and running the model.

The Data Preparation FW is a vertical FW that includes any specific data processing features that may be required by the application either in the preprocessing or the post-processing stage. This FW isolates and encapsulates miscellaneous functionalities instead of mixing them in with the previously defined FWs. Examples of such data preparation functionalities include parameter derivation and reordering or sorting.

By implementing these Frameworks using standard technologies, the resulting systems can be constructed as loosely coupled modules with a high degree of adaptability. In each new system, only the FWs needed for the specific application have to be customized, thereby reducing the complexity of the implementation, and hence its development and maintenance costs.

Framework Integration

The MIS FWs discussed in the previous section will be fully integrated into the system to enable users to set up and run the models, and visualize the results that will help in ocean monitoring and forecasting. The FWs identified here are designed with integration to the other FWs in mind. We expect integrating these FWs to be cost effective. Further, since the FW interfaces are well defined they are capable of being used within other applications. Thus, the FWs for format conversion, resolution conversion, and so on, can easily be used to develop other geospatial systems. However, building a system based on the sophisticated integration of independent FWs with formally defined interfaces creates a number of challenges, and this has been addressed in Mattson and Bosch (1997). Java- and XML-related technologies will be deployed in the implementation of the pilot MIS.

Integration of Legacy Systems

The environment domain and especially ocean modeling systems contain large amounts of software implemented in C and Fortran programming languages that are very difficult to reengineer. Existing or legacy software will be used as is in most cases, as this is code that has been tested, debugged, and used over several years. The various ways in which the legacy code can be used within the MIS include: (1) calling native methods from Java, using the JNI API; (2) encapsulating the entire executable application in a wrapper object; (3) remodularizing the legacy system to separate duplicated and redundant functionality from the core of the system; (4) reengineering simple functionality within legacy systems, such as selecting a geographic region; and (5) calling the executable directly from Java using the *Runtime* utility class. Hence, there are a number of options suited for integration of different types of legacy systems.

Related Work

Productivity and quality gains have been documented from several OO IS that have been developed and implemented in the recent past. Within the environment domain, various applications, many of them originally implemented in non-OO programming languages, have been successfully implemented in the OO paradigm. Reichert (1997) concluded that OO program design techniques helped make the program more robust, clarified program structure, and shortened development time. Furia and others (1995) concluded that the OO paradigm was particularly suitable to describing ecological models. The authors found that the properties of inheritance and dynamic binding were particularly useful in their application. They also recorded noticeable improvement in productivity, as the code was simpler to modify and more compact. Koesmarno (1997) documents that the OO paradigm was selected because of its versatility for a wide range of applications within the domain.

The Dynamic Environmental Effect Model (DEEM) is a software used to support modeling of terrestrial, aquatic, and atmospheric processes (arg95 1995). The system is very similar to the MIS outlined here in that it is object based, integrates various models and applications, and provides for flexibility and interoperability. While the DEEM (designed as early as 1993) is a monolithic reusable application, the MIS is built up of smaller, reusable object-oriented FWs that make up various aspects of the information system.

While none of the applications mentioned in this section used the FW technique, Mamrak and Sinha (1999) have documented reduced development and debugging time, as well as decrease in the cost of product development and an improvement in the quality of new applications in using the FW technique for redeveloping a medical IS.

Conclusion

Reusability is the keyword in designing and developing the MIS. By introducing reusability effectively at the design stage itself, we hope to produce Frameworks that are of high quality and that are designed to maximize reusability. The key features of the DIADEM MIS are as follows.

- The MIS is an object-oriented information system that accesses legacy systems.
- The MIS is designed using OO FWs that encapsulate some aspects of the MIS. Each of these FWs can be extended and customized for a particular application scenario.
- The MIS is developed to operate under the UNIX operating system and to operate in a "mixed" language environment. The core MIS software utilizes Java. Numerical models and native methods are run in whatever language in which they were developed, for example C or Fortran.
- The MIS makes use of off-the-shelf Frameworks, tools, and software wherever possible—such as for data management and GUI features.

We also summarize some of the technical, economical, and practical reasons why ISs in the environmental domain should move to OOT.

- The current trend is towards distributed applications. Traditional software has not been developed with distribution in mind.
- There is a definite shift from rigid character-based to flexible graphical and networked user interfaces.
- There is a need to process new data rapidly, to integrate new algorithms more quickly, and to come up with predictions and forecasts faster than before. This "bigger appetite" effect indicates a need for a new technology such as OO, which provides an environment for rapid application development and a means to feed that appetite.
- Reuse of design and code, as opposed to code alone, promotes the development of reliable software with relatively lower development and maintenance costs.
- Automated and semiautomated software help to flatten the learning curve and hence increase software use.
- OOT makes it possible to continue working with legacy code, while developing the new software. Thus software developed as legacy code can be maintained and scientists can continue to work in an environment they are used to.

In this article we have examined the requirements of a potential MIS by studying the requirements of the DIADEM ocean monitoring and prediction system. The diversity and complexity of marine data can be taken care of, to a large extent, by using the object-oriented approach. By using the DIADEM ocean monitoring and forecasting system as the first application for developing a FW for the marine environment, all major FWs required have been identified. Similar studies of the underwater Acoustic Information System (AIS) and the Ground Water Information System (GWIS) have been undertaken. This has led to the identification of similar and dissimilar aspects of information systems within the environment domain, which in turn has led to the identification of the FWs (Jacob 1999). Basing the design and implementation of these FWs on standard object-oriented technologies and methods will promote reuse of these components in new systems within and across domains.

We are currently working on developing the individual FWs identified in Figure 8. A pilot version of the horizontal Format Conversion FW was developed and tested for data from different application scenarios within the environment domain. Work is now focussed on developing the vertical Simulation Model FW. The experience gained should provide insight into the differences in developing the horizontal and vertical FWs, and also into integration of the FWs.

Issues pertaining to data management, data distribution, and remote access still need to be addressed. However, we believe that the OO analysis and design presented here are a step in the right direction for implementing the DIADEM MIS.

References

Adair, D. 1995. Introducing Object Oriented Frameworks, Taligent Inc. http://www.developer.ibm.com/library/aixpert/feb95/aixpert_feb95/boof.html and http://www.ibm.com/java/education/oobuilding/index.html [online].

Anon. 1998a. Bathymetry data, Terrainbase project: National Geographic Data Center and World Data Center. ftp://ncardata.ucar.edu/datasets/ds759.2/ [online].

Anon. 1998b. Information, support and software from the HDF group at NCSA. The NCSA HDF home page: http://hdf.ncsa.uiuc.edu/index.html [online].

arg95. 1995. The dynamic information architecture system: A high level architecture for modeling and simulation. White paper prepared by the Decision and Information Sciences Division, Argonne National Laboratory. http://www.dis.anl.gov/DEEM/DIAS/diaswp.html [online].

Bentsen, M., G. Evensen, H. Drange, and A. D. Jenkins. 1999. Coordinate transformation on a sphere using a conformal mapping. *Monthly Weather Review* 127:2733–2740.

Bleck, R., C. Rooth, D. Hu, and L. T. Smith. 1992. Salinity-driven thermohaline transients in a wind- and thermohaline-forced isopycnic coordinate model of the North Atlantic. *J. Phys. Oceanogr.* 22:1486–1515.

Bleck, R. 2002. An oceanic general circulation model framed in hybrid isopycnic-Cartesian coordinates. *Ocean Modeling* 4:55–88.

Buschmann, F., R. Meunier, H. Rohnert, P. Sommerland, and M. Stal. 1996. Pattern-oriented software architecture: A system of patterns. New York, NY: Wiley.

Demeyer, S., T. D. Meijler, O. Nierstrasz, and P. Steyaert. 1997. Design guidelines for tailorable frameworks. *Communications of the ACM* 40(10):60–64.

Drange, H. 1994. An isopycnic coordinate carbon cycle model for the North Atlantic: And the possibility of disposing of fossil fuel CO2 in the ocean. PhD thesis, Univ. of Bergen/NERSC, Bergen, Norway.

Evensen, G. 1992. Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model. *J. Geophys. Res.* 98:17905–17924.

Evensen, G. 1994. Inverse methods and data assimilation in nonlinear ocean models. *Physica D* 77:108–129.

Evensen, G. 1998. Development of advanced data assimilation systems for operational monitoring and forecasting of the North Atlantic and Nordic Seas. EC MAST-III Project. http://diadem.nersc.no

Fasham, M. J. R., H. W. Ducklow, and S. M. McKelvie. 1990. A nitrogen-based model of phytoplankton dynamics in the oceanic mixed layer. *J. Marine Res.* 48:591–639.

Furia, L. D., A. Rizzoli, and R. Arditi. 1995. Lakemaker: A general object oriented software tool for modelling the eutrophication process in lakes. *Environmental Software* 10(1):43–64.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1995. Design patterns: Elements of reusable object-oriented software. Reading, MA: Addison-Wesley Longman.

Geary, D. M. 1999. *Graphic Java 2, Vol 2. Swing*. Palo Alto, CA: Prentice Hall PTR.

Hamre, T. 1995. Development of semantic spatio-temporal data models for integration of remote sensing and *in-situ data* in a marine information system (MIS). Ph.D. thesis, University of Bergen/NERSC, Bergen, Norway.

Jacob, A. 1999. Development of Object Oriented Frameworks for spatio-temporal information systems. http://www.nersc.no/~torill/anita_jacob/ICSE99/onlineabs.html [online].

jai. 2002. Java Advanced Imaging API. SUN Microsystems Ltd. http://java.sun.com/products/java-media/jai/index.html [online].

Jones, P. W. 1999. First- and second-order conservative remapping schemes for grids in spherical coordinates. *Monthly Weather Review* 127(9):2204–2210.

Koesmarno, H. K. 1997. OFSIM: A simulation shell for forest management. *Environmental Modelling and Software* 12(1):75–83.

Larman, C. 2001. Applying UML and Patterns: *An Introduction to Object-Oriented Analysis and Design and the Unified Process*, 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR.

Legates, D. R., and C. J. Wilmott. 1990. Mean seasonal and spatial variability in gauge corrected, global precipitation. *Int. J. of Climatol.* (10):111–127.

Liang, S. 1999. Java™ Native Interface: Programmer's guide and specification. Palo Alto, CA: Addison-Wesley.

Mamrak, S. A., and S. Sinha. 1999. A case study: Productivity and quality gains using object oriented framework. *Software—Practice and Experience* 29(6):501–518.

Mattsson, M., and J. Bosch. 1997. Frameworks composition: Problems, causes and solutions. TOOLS'97, USA.

McClain, E. P., W. G. Pichel, C. C. Walton, Z. Ahmad, and J. Sutton. 1983. Multichannel improvements to satellite-derived global sea surface temperatures. *Adv. Space Res.* 2(6):43–47.

Reichert, P. 1997. Design techniques of a computer program for the identification of processes and the simulation of water quality in aquatic systems. *Environmental Software* 12(3):199–210.

Rew, R., G. Davis, S. Emmerson, and H. Davies. 1996. NetCDF User's guide, ver 2.4: An interface for data access. http://ibmbb2.aug.ipp-garching.mpg.de:8001/~dpc/netcdf_guide/guide_toc.html [online].

Rizzoli, A. E., and W. J. Young. 1997. Delivering environmental decision support systems: Software tools and techniques. *Environmental Modelling and Software* 12(2–3):237–249.

Rogers, G. F. 1997. Framework-based software development in C++. Upper Saddle River, NJ: Prentice Hall PTR.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and William Lorensen. 1991. Object-oriented modeling and design. Edgewood Cliffs, NJ: Prentice Hall.

Rumbaugh, J., I. Jacobsen, and G. Booch. 1999. *The Unified modelling language reference manual*. Reading, MA: Addison-Wesley.

Sarmiento, J. L., R. D. Slater, M. J. R. Fasham, H. W. Ducklow, J. R. Toggweiler, and G. T. Evans. 1993. Seasonal three-dimensional ecosystem model of nitrogen cycling in the North Atlantic euphotic zone. *Global Biogeochem. Cycles* 7:417–450.

Schmid, H. A. 1997. Systematic frameworks design by generalization. *Communications of the ACM* 40(10):48–51.

Szyperski, C. 1998. Components software: Beyond OO programming. New York, NY: Addison-Wesley.